

ABC

N° 20

COURS
D'INFORMATIQUE
PRATIQUE
ET FAMILIALE

INFORMATIQUE



Histoires d'ordinateur

RPP Connexion

Le Memotech MTX 512

Le sixième sens

EDITIONS
ATLAS

M 6062-20-12F

85 FB-3,80 FS-\$1.95

Dans toutes les librairies



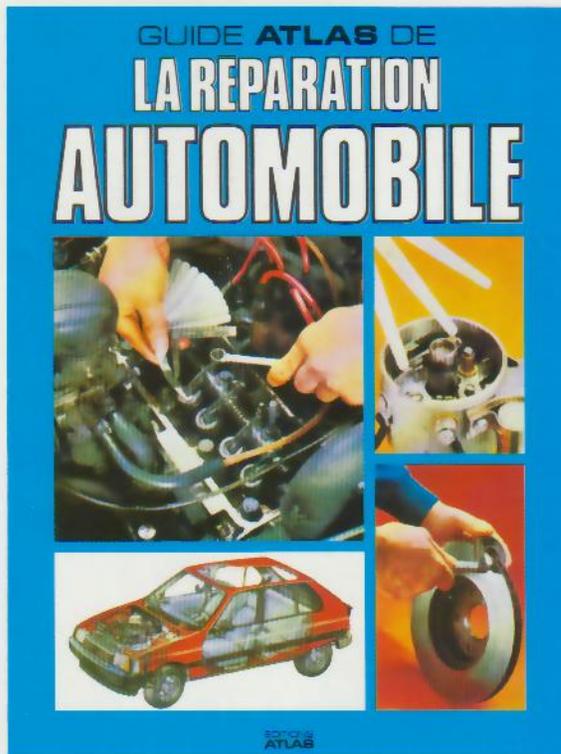
Les Grands Détectives

Sept enquêtes originales

Confrontés aux énigmes les plus diaboliques de la littérature policière, les Grands Détectives ont toujours suscité admiration, passion et mystère. Sept enquêtes originales où humour nostalgique et pastiche subtil se complètent à merveille, prestigieusement illustrées par un maître du genre, contribuent à faire revivre les plus Grands Détectives, ces indispensables compagnons de nos longues et envoûtantes lectures nocturnes...

*Un volume relié.
Jaquette illustrée. 144 pages.
80 illustrations originales
de Tom Adams.
Format : 21,5 x 28 cm.*

Dans toutes les librairies

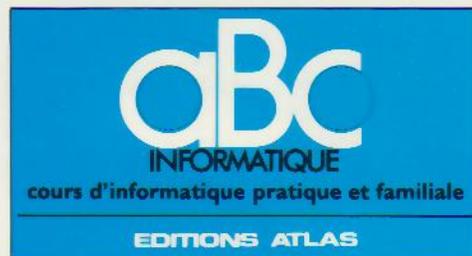


Guide Atlas de la réparation automobile

Le seul moyen de faire durer et de maintenir sa voiture dans de bonnes conditions d'utilisation, c'est encore de l'entretenir et de la réparer soi-même.

Véritable encyclopédie de l'automobile, ce guide pratique d'entretien et de réparation permet, en n'ayant qu'un minimum de matériel et de connaissances de se familiariser avec les secrets de la mécanique automobile.

*Un volume relié,
sous couverture illustrée.
240 pages.
800 illustrations en noir et
blanc et en couleurs.
Format : 22 x 29 cm.*



Édité par EDITIONS ATLAS s.a., tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Belgique : EDITIONS ATLEN s.a., Bruxelles.

Canada : EDITIONS ATLAS CANADA Ltée, Montréal Nord

Suisse : FINABUCH s.a., EDITIONS TRANSALPINES, Mezzovico.

Realise par EDENA s.a., 29, boulevard Edgar-Quinet, 75014 Paris. Tél. : 320-15-01.

Direction éditoriale : J.-Fr. Gautier, Service technique et artistique : F. Givone et J.-Cl. Bernar. Iconographie : J. Pierre. Correction : B. Noël.

Publicité : Anne Cayla. Tél. : 202-09-80.

VENTE

Les numéros parus peuvent être obtenus chez les marchands de journaux ou, à défaut, chez les éditeurs, au prix en vigueur au moment de la commande. Ils resteront en principe disponibles pendant six mois après la parution du dernier fascicule de la série (Pour toute commande par lettre, joindre à votre courrier le règlement, majoré de 10 % de frais de port.)

Pour la France, s'adresser à EDITIONS ATLAS, tour Maine-Montparnasse, 33, avenue du Maine, 75755 Paris Cedex 15. Tél. : 538-52-70.

Pour les autres pays, s'adresser aux éditeurs indiqués ci-dessous.

SOUSCRIPTION

Les lecteurs désirant souscrire à l'ensemble de cet ouvrage peuvent s'adresser à :

France : DIFFUSION ATLAS, 3, rue de la Tave, 28110 Lucé. Tél. : (37) 35 40-23.

Belgique : EDITIONS ATLEN s.a., 55, avenue Huart-Hamoir, 1030 Bruxelles. Tél. : (02) 242-39-00. Banque Bruxelles-Lambert, compte n° 310-0018465-24 Bruxelles.

Canada : EDITIONS ATLAS CANADA Ltée, 11450 boulevard Albert-Hudon, Montréal Nord, H 1G 3J9

Suisse : FINABUCH s.a., EDITIONS TRANSALPINES, zona industriale 6849 Mezzovico-Lugano. Tél. : (091) 95-27-44.

RELIEZ VOS FASCICULES

Des reliures mobiles, permettant de relier 12 fascicules, seront en vente en permanence chez votre marchand de journaux.

ATTENTION : ces reliures, présentées sans numérotation, sont valables indifféremment pour tous les volumes de votre collection. Vous les numéroterez vous-même à l'aide du decalque qui est fourni (avec les instructions nécessaires) dans chaque reliure.

En vente tous les vendredis. Volume II, n° 20.

ABC INFORMATIQUE est réalisé avec la collaboration de Trystan Mordrel (secrétariat de rédaction), J.-P. Bourcier (coordination), P. Bazin, J.-P. Moulon, Cl. Rémy (traduction), Ghislaine Goullier (fabrication), Marie-Claire Jacquet (iconographie), Patrick Boman (correction). Crédit photographique, couverture : Jean Riby - Matra-Hachette.

Directeur de la publication : Paul Bernabeu. Imprimé en Italie par I.G.D.A. Officine Grafiche, Novara. Distribution en France : N.M.P.P. Tax. Dépôt légal : mar 1984. 18845. Dépôt légal en Belgique : D/84/2783/27.

© Orbis Publishing Ltd., London.
© Editions Atlas, Paris, 1984.

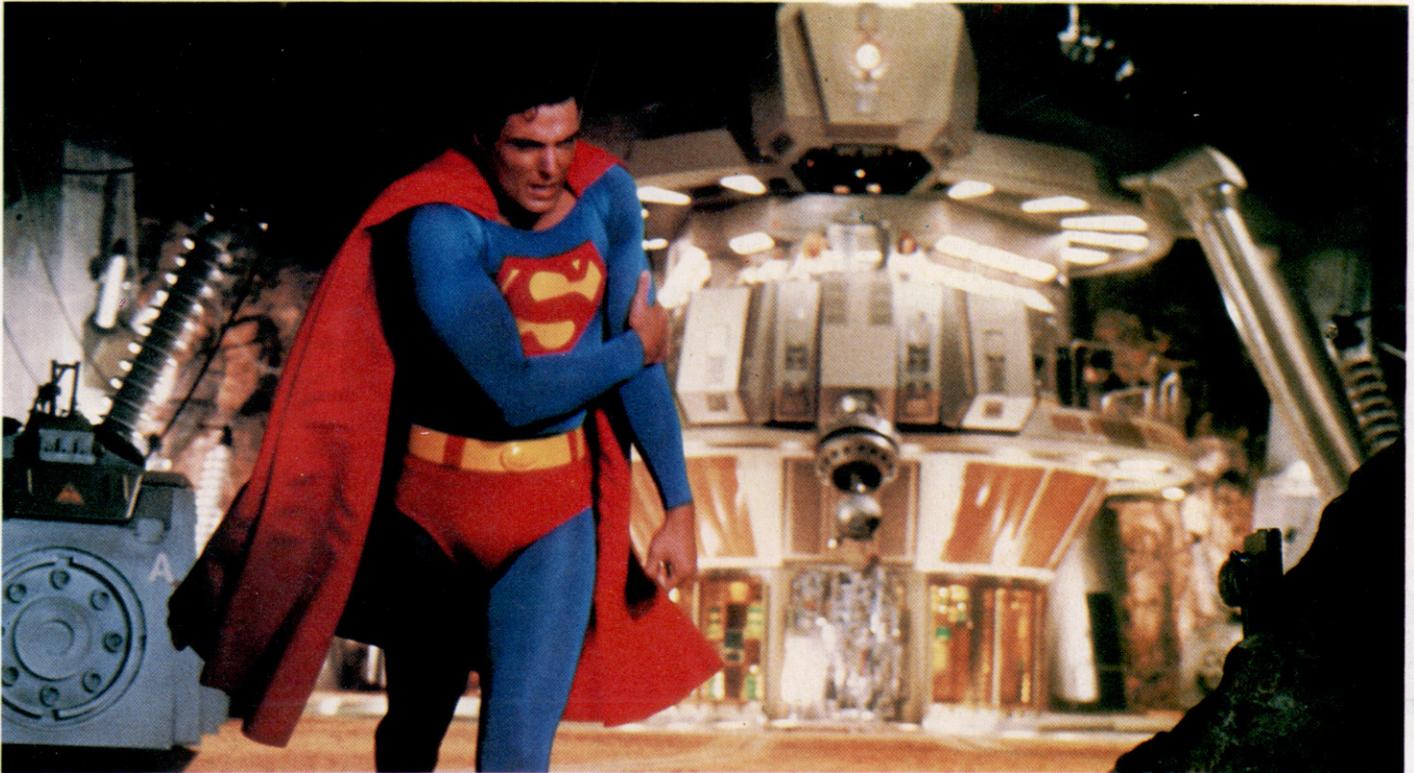
A NOS LECTEURS

En achetant chaque semaine votre fascicule chez le même marchand de journaux, vous serez certain d'être immédiatement servi, en nous facilitant la précision de la distribution. Nous vous en remercions d'avance.

Les Editions Atlas



Histoires d'ordinateur



Les ordinateurs sont souvent les héros de la science-fiction. Les auteurs ne se sont guère trompés dans les descriptions technologiques.

Baucoup de réalisations scientifiques ou techniques ont été décrites dans des livres de fiction ou des films, bien avant qu'elles n'existent réellement. Arthur C. Clarke, l'auteur de *2001 : L'Odyssée de l'espace*, fut le premier à décrire dans un magazine du début des années cinquante l'existence d'un satellite géostationnaire, soit près de vingt années avant leur véritable existence dans l'espace. Dans une nouvelle, Robert Heinlein explique le rôle d'un manipulateur contrôlé à distance alors que les bras robots sont encore loin d'apparaître sur le marché. Beaucoup d'inventeurs et de techniciens se sont réellement inspirés de l'imagination inventive des écrivains de science-fiction et des réalisations de films.

Ces ordinateurs de fiction n'ont, cependant, que peu de ressemblance avec la réalité. Dans le film futuriste *Rollerball*, par exemple, l'ordinateur qui possède l'ouïe et la parole a la forme d'un réservoir cubique pour liquide. Naturellement, les ordinateurs ressemblant à une chose réelle enlèvent toute tension dramatique et sont donc moins intéressants. Mais il ne fait aucun doute qu'au cours des années soixante et

soixante-dix, les films présentant des ordinateurs proches, dans leur forme, des machines actuelles ont largement contribué à éduquer le grand public en démythifiant ces nouveaux produits.

Il ne fallut pas attendre longtemps pour que l'imagination créative de certains emboîte le pas de Charles Babbage et de ses travaux sur une machine analytique, au milieu du XIX^e siècle. En 1879, Edward Page Mitchell écrit une histoire qui raconte comment une machine à calculer implantée dans le cerveau d'un simple d'esprit transforme celui-ci en génie. Sur plusieurs points, cet auteur décrivait correctement l'avenir. D'abord, il introduisait l'idée de miniaturisation : sa machine était suffisamment petite pour se loger dans un crâne et assez puissante pour agir sur l'intelligence. Ensuite, Mitchell donnait une préfiguration de ce qu'allait être l'interconnexion entre l'ordinateur et le corps humain. Aujourd'hui, plus d'un siècle après cette histoire, ces techniques sont en train d'être mises au point et sont testées sur le système nerveux.

Généralement, peu d'écrivains possèdent une grande connaissance de l'architecture des ordinateurs, même si nombre d'entre eux sont ingénieurs et utilisent des ordinateurs dans leur travail. Mais, est-ce gênant ? Il n'existe pas de réelles difficultés pour représenter, par l'image, des voyages intergalactiques même si leurs créateurs ne sont pas des astrophysiciens très qualifiés ou des experts en balistique. De la même façon, il n'y a aucune raison pour que les écri-

Superman III

La fraude à l'ordinateur est le thème central du film *Superman III*. Richard Pryor joue le rôle d'un gangster qui établit sa fortune en dérobant 50 % de toutes les transactions qui passent par l'ordinateur d'une banque. Le sujet est fondé sur des cas réels de fraude de ce genre. Le film se termine par la destruction du plus grand ordinateur du monde ayant été construit uniquement dans ce but criminel. (Cl. Columbia Warner.)



vains ne puissent spéculer sur les possibilités des futures générations d'ordinateurs sans en avoir une connaissance profonde.

Cette projection sur l'avenir a conduit, cependant, à une espèce de standardisation des capacités des ordinateurs dans toute la science-fiction. Ainsi, ce genre d'ordinateur doit avoir enregistré dans sa mémoire toutes les informations et les idées possibles pour les retraduire instantanément, de la même manière que le fait un être humain. L'ordinateur HAL de 2001 : *L'Odyssée de l'espace* est l'exemple typique

film *Dark Star*, par exemple, un ordinateur-contrôleur de bombe présente toutes les caractéristiques d'un tueur psychotique. Ainsi décrits, ces super-ordinateurs appartiennent au royaume de la fantaisie. Mais admettons que l'on peut reconnaître dans les ordinateurs d'aujourd'hui les « futurs ancêtres » de ceux que nous venons de décrire.

Les grandes capacités de mémoire avec de très courts temps d'accès sont déjà du domaine réel. Dès le début des années quatre-vingt, les mémoires de 1 milliard d'octets apparaissent et les machines professionnelles les plus rapides traitent les informations à plus de 10 millions d'instructions par seconde. Dans le domaine de la parole, nous sommes très près d'atteindre la perfection des ordinateurs montrés dans les films. La qualité de la parole ne dépend que de la quantité de mémoire qui lui est consacrée et de la vitesse de traitement. La reconnaissance de la voix soulève d'autres problèmes, plus délicats. Deux personnes peuvent parler le même langage, l'ordinateur aura encore beaucoup de difficultés pour les reconnaître.

La reconnaissance des formes en est également à ses débuts, mais les progrès sont rapides. Les robots industriels sont déjà capables de réaliser des prouesses et choisir des pièces désirées parmi d'autres. Tout l'ensemble du problème de la reconnaissance visuelle dépend de l'étendue des instructions — du vocabulaire — qu'on lui consacre; il s'agit encore d'une question de capacité mémoire et de vitesse de traitement. En ce qui concerne la synthèse alimentaire, le problème présente deux aspects très différents: il n'est pas encore possible d'aboutir à des produits aussi ressemblants que la viande ou des frites, mais on sait en faire qui en ont le goût et l'odeur. Fermez les yeux!

Tous les auteurs ne vont pas jusqu'à donner à leur ordinateur une image très proche de l'être humain. John Brunner, par exemple, décrit dans une de ses nouvelles de science-fiction publiée en 1969 le monde de 2010, époque où selon lui la surpopulation et la famine arrivent au point limite. Son ordinateur possède bien une mémoire considérable et une vitesse de traitement incroyable (il est directement branché avec toutes les télévisions de la terre) mais son langage d'interrogation reste très voisin de ceux que l'on utilise aujourd'hui :

PROGRAMME REJETÉ

Q raison du rejet

ANOMALIES DANS DONNÉES DE BASE

Q définissez Q précisez

DONNÉES DANS CATEGORIES SUIVANTES NON ACCEPTABLES

HISTOIRE COMMERCE SOCIAL ET CULTURE

Q acceptez les données telles quelles

QUESTION SANS SIGNIFICATION ET INOPÉRANTE



d'une telle machine. Généralement, ces super-ordinateurs, imaginés par les écrivains de science-fiction, sont omniprésents et d'une manière telle que chaque utilisateur a l'impression d'être le seul interlocuteur. Une sortie vocale (sans trop montrer qu'il pourrait s'agir de simples phonèmes ficelés ensemble) et une reconnaissance de la voix sont absolument essentielles dans la description de ces machines. Mais la reconnaissance visuelle des objets et la capacité de synthétiser les aliments (peut-être à partir de sa constitution basique élémentaire) font partie des attributs acceptables de l'ordinateur.

Les super-ordinateurs que nous venons de décrire sont également animés par de nombreux autres facteurs humains, au point d'apparaître comme des êtres supérieurs. Toutefois, l'ordinateur peut tomber dans le mauvais genre, la malveillance, et parfois il se détraque. Dans le

L'auteur a donc préféré privilégier un langage que les utilisateurs d'ordinateur trouveront proche de ce qu'ils connaissent. Dans ce contexte, il n'est pas surprenant que cette nouvelle ait connu un très grand succès public.

Bizzarrerie spatiale

HAL, l'ordinateur d'Arthur C. Clarke dans le film 2001 : *L'Odyssée de l'espace* est l'exemple typique de l'omniprésence de l'ordinateur dans les films de science-fiction. HAL connaît un certain nombre d'informations que les membres de l'équipage ne connaissent pas. Cela entraîne la machine à croire que l'homme n'est pas indispensable. On sait que Clarke choisit les lettres HAL, car elles précèdent respectivement celles d'IBM dans l'alphabet. (Cl. B.F.I. Stills.)

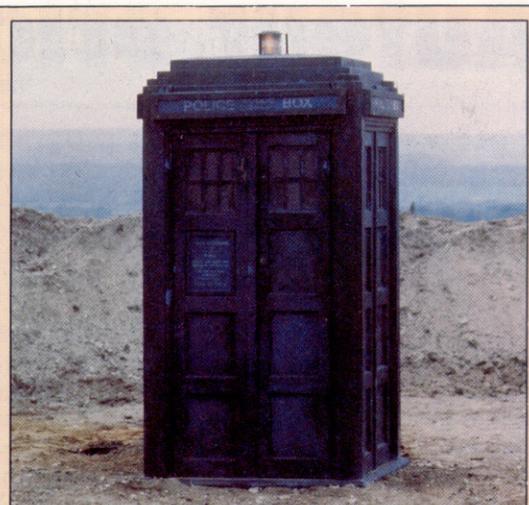


Dans les productions littéraires et filmiques les plus récentes, l'ordinateur n'est plus seulement un élément du décor, il devient le décor à lui seul. L'exemple typique est le film *Tron* de Walt Disney. Nous avons déjà évoqué l'exceptionnelle animation de ce film qui fait la part belle à la fois au monde réel et au squelette de l'ordinateur. Le monde extérieur apparaît comme l'espace des hommes de la programmation, des logiciels, des informaticiens. A l'intérieur de la machine, les caractéristiques du système d'exploitation, des structures des programmes dominant et l'architecture de l'ordinateur provoque le scénario où l'action va se dérouler. Rappelons que *Tron* (le titre du film) est dérivé d'un système d'exploitation mnémotechnique intitulé : Trace ON, en anglais.

D'autres produits de pure fiction ne font pas réellement mention des ordinateurs, mais le lecteur n'a aucun doute quant à l'importance de la puissance de ces machines sans laquelle l'action ne pourrait se dérouler. Dans ce domaine, on peut penser à George Orwell et Aldous Huxley. *1984* et *Le Meilleur des mondes* montrent des sociétés totalement sous la dépendance d'une petite minorité. Ces deux livres sont certainement à l'origine d'une certaine angoisse à l'encontre de la force des ordinateurs.

Il ne nous est pas possible d'être réellement exhaustif dans cette rétrospective sur l'usage de l'ordinateur dans la fiction. Mais ajoutons que, parfois, l'ordinateur n'est que le prétexte à création de situations dans lesquelles le héros devra se débattre.

En fait, il ne faut pas oublier que l'écriture sur les ordinateurs ne relève pas que du domaine de la fiction. Pour des milliers de spécialistes, les livres réellement d'informatique peuvent connaître autant de succès que ceux de fiction si



Pense-bête

Moitié outil de transmission, moitié intelligence artificielle, l'ordinateur n'a de limite que celle de l'imagination des auteurs. Mais la conversation doit passer par le clavier et l'écran, preuve de l'intelligence limitée de la machine, comme de l'étendue de sa mémoire. (Cl. BBC Stills.)

l'écriture est captivante. Cela est déjà arrivé. Mais même si l'histoire est celle des hommes créateurs de l'informatique, la star reste toujours l'ordinateur lui-même. Et demain, cela sera encore plus vrai. En effet, il sera possible de faire revivre dans de nouvelles situations, sur les écrans, des acteurs célèbres en profitant de l'énorme puissance de la mémoire des ordinateurs. Comment? Grâce à la création d'images à partir de celles déjà existantes. Les visages et les gestes sont déjà stockés!



War Games

Inconsciemment, un adolescent passionné d'informatique s'est branché sur le réseau de défense de l'OTAN, en cherchant à communiquer avec un ami sans passer par le réseau public du téléphone. Croyant que les images qu'il reçoit ne sont qu'un jeu, il se met à jouer jusqu'au moment où il découvre qu'il est à l'origine de la Troisième Guerre mondiale... (Cl. United Artists.)

Auteur original

Il est possible d'écrire des programmes qui généreront d'autres programmes ou corrigeront des erreurs.

« Si les ordinateurs sont si malins, à quoi bon des humains pour les programmer ? » Les informaticiens chevronnés hausseront les épaules devant cette question, mais elle n'est pas aussi bête qu'elle le semble. De nombreuses recherches sont entreprises pour écrire des programmes pouvant générer d'autres programmes, et des systèmes d'exploitation capables de corriger les erreurs humaines de codage sont mis au point.

Considérons le message ERREUR de SYNTAXE? fréquemment rencontré par les utilisateurs d'ordinateurs domestiques. On peut être furieux que ce message donne si peu d'informations. Avec un gros système informatique, un compilateur donnera bien plus d'indications

quant à la nature de l'erreur rencontrée. Par exemple, voici un message d'erreur :

```
1090 LET A = (C*2 + F$)* ((FG - C)* TH + 1))
```

ERREURS : 1) VARIABLE DE CHAÎNE F\$ NON PERMISE
2) DERNIÈRE PARENTHÈSE EN TROP

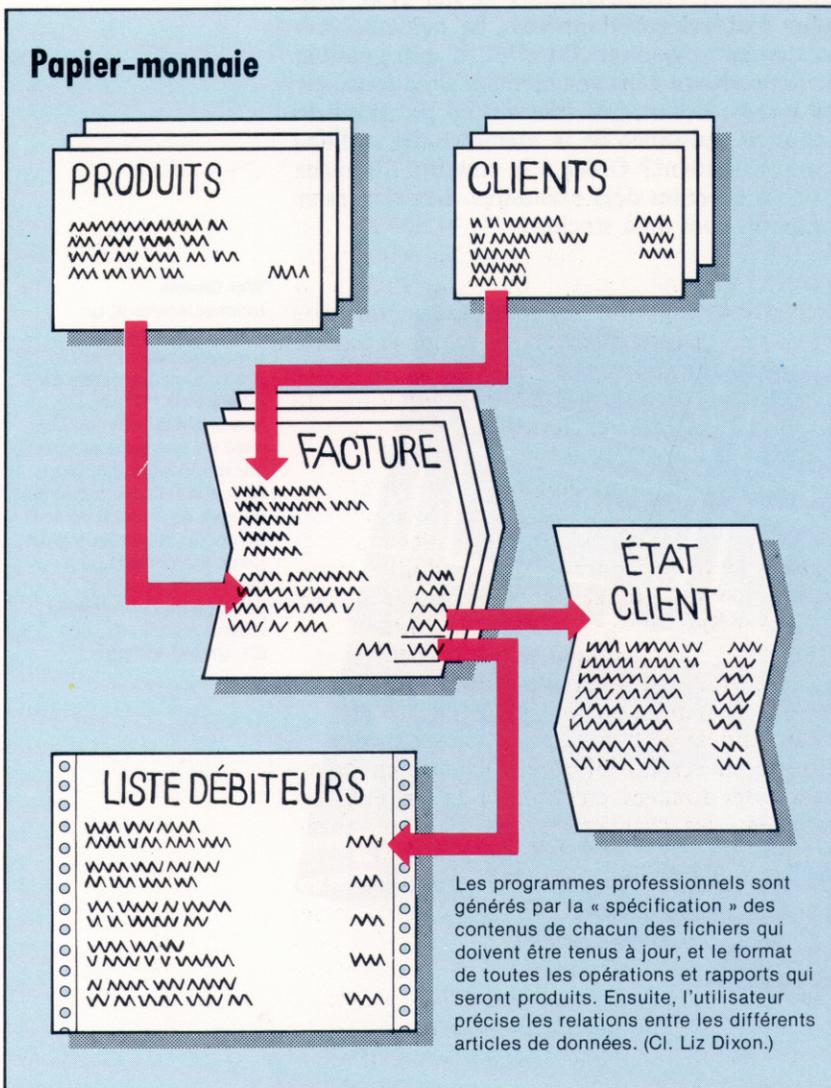
Il n'y a en principe aucune raison pour que de telles techniques ne puissent pas être utilisées dans un interpréteur d'ordinateur domestique — le coût de la mémoire morte supplémentaire que nécessiterait le stockage des programmes serait minimal. Mais peu d'ordinateurs emploient des procédures de contrôle d'erreur même superficielles. Cependant, il est souvent possible d'acheter une ROM supplémentaire ou des cartouches de logiciel enfichables pour étendre l'ensemble des commandes BASIC disponibles, en particulier celles relatives aux erreurs de programmes. Ces commandes BASIC comprennent :

HELP — affiche la ligne de programme et met en évidence la position du caractère où s'est arrêtée l'exécution du programme. Cela indique généralement, mais pas toujours, l'origine de l'erreur de syntaxe.

DUMP — affiche une liste de tous les noms de variables et leurs contenus couramment utilisés par le programme. Cela permet de déduire jusqu'où est allé le programme avant de rencontrer l'erreur.

TRACE — affiche (dans une fenêtre au coin de l'écran) le (ou les) numéro(s) de ligne exécuté(s) pendant que le programme tourne. L'utilisateur peut ainsi suivre le mouvement du programme et s'assurer, entre autres, que les sous-programmes sont exécutés dans l'ordre désiré.

Écrire des programmes qui permettent à un ordinateur de corriger les erreurs humaines de codages n'est généralement pas simple. Mais, dans certains cas, c'est vraiment facile. Par exemple, nous savons que toutes les lignes de programme doivent commencer par un mot clé BASIC (quoique certaines machines permettent d'omettre le mot LET). De ce fait, si une ligne commence par PRUNT ou PRONT, il sera facile de trouver qu'il devait s'agir de PRINT. Dans le cours de programmation en BASIC, nous avons discuté de l'idée d'assortiments flous (algorithmes pour trouver ce qui se rapproche le plus d'une expression) applicables aux mots clés de programmes. D'une autre manière, l'interpréteur pourrait simplement comporter une liste d'erreurs de frappe communes et leurs équivalents corrects. Par mesure de sécurité, il vau-



draît mieux, pour l'ordinateur, vérifier toutes les altérations faites avec l'opérateur.

Mais en dehors de ces procédures, la correction automatique présente beaucoup plus de difficultés. Dans l'exemple que nous avons donné, est-ce que F\$ est une faute d'impression pour F, FS, F4, ou quelque chose d'entièrement différent? Si vous montriez tout le listage à un autre programmeur compétent, il pourrait identifier les fautes et faire les corrections. Il se servirait de deux critères dans sa décision : le contexte dans lequel la ligne de programme apparaît et sa propre expérience.

Curieusement, cette technique a été plus largement appliquée pour corriger un texte que pour vérifier un langage de programmation. Un bloc logique d'orthographe, par exemple, passera en revue un texte et mettra en évidence tous les mots qui ne correspondront pas aux entrées de son dictionnaire, soit peut-être 50 000 mots stockés sur disque. La plupart de ces blocs ont la possibilité d'apprendre de nouveaux mots (comme l'orthographe de noms propres) et de les ajouter à leur répertoire. Les plus sophistiqués suggèrent même l'orthographe correcte si une similitude est détectée. On a également développé expérimentalement des processeurs de mots qui peuvent appliquer le même processus à la grammaire et au style — en relevant les erreurs de ponctuation, les répétitions de mots dans un paragraphe, les métaphores confuses et les adjectifs impropres. Ceux-ci fonctionnent également en examinant le contexte de chaque expression par référence à une bibliothèque d'exemples précédemment utilisés.

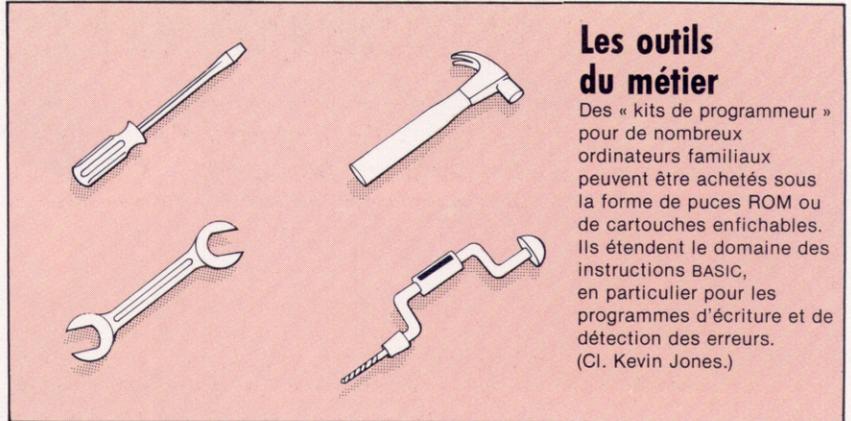
Il a cependant fallu plus d'efforts pour développer des systèmes capables de créer des programmes, plutôt que de corriger ceux qui existaient déjà. En 1981, on annonça un logiciel qui déclencha l'une des plus âpres batailles de l'industrie micro-informatique. Dénommé « Le dernier », il avait la prétention de pouvoir écrire n'importe quel programme. Cette prétention était injustifiée, mais « Le dernier » contribua utilement à développer certains types de programmes, essentiellement professionnels. Il existe maintenant plusieurs produits de ce genre sur le marché des micro-ordinateurs professionnels et pour quelques ordinateurs familiaux. Ils sont regroupés sous le nom de « générateurs de programmes ».

Considérons à présent les concepts fondamentaux sous-jacents à un programme capable d'écrire des programmes. Voici un exemple banal :

```
10 PRINT « QUE VOULEZ-VOUS QUE LE PROGRAMME
AFFICHE SUR L'ÉCRAN ? »
20 INPUT A$
30 PRINT « LE PROGRAMME EST : »
40 PRINT « 10 PRINT », CHR$(34); A$; CHR$(34)
```

Si vous répondez BONJOUR à la question, le programme (qui devrait tourner sur la plupart des ordinateurs familiaux) doit afficher la ligne :

```
LE PROGRAMME EST :
10 PRINT « BONJOUR »
```



Les outils du métier

Des « kits de programmeur » pour de nombreux ordinateurs familiaux peuvent être achetés sous la forme de puces ROM ou de cartouches enfichables. Ils étendent le domaine des instructions BASIC, en particulier pour les programmes d'écriture et de détection des erreurs. (Cl. Kevin Jones.)

En appliquant la même technique à tous les stades d'une application, vous pouvez écrire un générateur de programme très simple. Si toutes les questions posées sont simplement exprimées, il sera possible à un néophyte de développer un programme simple utilisant votre générateur.

Les générateurs de programmes commercialisés utilisent les mêmes techniques. La plupart des applications professionnelles consistent en une combinaison de cinq processus distincts (entrée des données, affichage sur l'écran ou l'imprimante, stockage dans un fichier de données, recherche de données et calcul).

Même si les générateurs de programmes deviennent plus performants, il est peu probable qu'ils remplacent les programmeurs humains dans l'avenir immédiat, car ils restent limités. Premièrement, la technique décrite convient fort bien pour des applications professionnelles de type commercial telles que la tenue de stock, mais généralement ces générateurs de programmes ne peuvent être appliqués pour écrire des programmes de traitement de texte ou de jeux. Deuxièmement, parce que le générateur de programmes doit utiliser ces sous-programmes standard, le listage obtenu ne sera pas aussi efficace (en termes de vitesse ou de mémoire) qu'il l'aurait été s'il avait été écrit exprès par un programmeur. Troisièmement, les programmes produits par des générateurs ne sont généralement pas aussi agréables à utiliser que les systèmes couramment produits par des programmeurs humains. Par exemple, ils font rarement bon usage des facultés graphiques offertes par les machines les plus récentes.

Enfin, les générateurs de programmes actuellement disponibles ne peuvent que remplacer l'étape finale de la programmation : le codage. Il reste encore à l'utilisateur la mise en forme exacte des données, en entrée et en sortie. Généralement, les premiers stades de la programmation sont les plus difficiles et exigent des compétences spécifiques. La plupart des grandes sociétés emploient des spécialistes appelés « analystes systèmes » pour préciser les programmes dont elles ont besoin, et ces précisions sont alors traduites en code par des programmeurs. Les générateurs de programmes ont encore à acquérir toutes les compétences nécessaires pour créer un programme d'ordinateur.



L'arc-en-ciel

Certains ordinateurs domestiques peuvent devenir très puissants grâce à des accessoires. Prenons l'exemple du Sinclair et poussons-le à ses limites.

Microdrive

Le Microdrive utilise des cartouches comportant une bande sans fin dont chaque point passe devant la tête de lecture/écriture toutes les sept secondes. Le transfert de l'information se fait au rythme de 6 K par seconde (quatre fois la vitesse d'une cassette normale). Il est possible de connecter jusqu'à huit Microdrives (700 K au moins).

Coupleur acoustique

Le Micro-Myte 60 montré ici permet à un ordinateur de communiquer avec un autre.

Clavier

Le clavier FDS de Fuller montré ici comporte des touches de fonction et une barre d'espacement.

Manette de jeux

Il est possible, via l'interface 2 de Sinclair, de brancher toutes manettes compatibles avec Atari, quel que soit leur mode de fonctionnement. Il est possible également de brancher deux manettes à la fois.

Cartouche RAM

Le plus petit Spectrum de 16 K peut être transformé par l'adjonction d'une cartouche RAM enfichable de 32 K.

Dès sa création, en 1982, le Spectrum de Sinclair réalisait une percée technologique et commerciale. Après une année de production, ses ventes (600 000 unités) représentaient la moitié du parc de micro-ordinateurs vendus en Angleterre. Une grande surprise, même chez Sinclair. Le Spectrum représente une nette amélioration par rapport au modèle précédent de Sinclair, le ZX-81 : 16 ou 48 K de mémoire vive en standard ; 8 couleurs pour la marge, le fond et le texte ; une capacité graphique de haute résolution ; un clavier amélioré mais encore imparfait ; la possibilité de générer des sons simples. Des fabricants indépendants ont profité du succès du Spectrum pour proposer des accessoires. Sinclair lui-même a pensé aux périphériques en créant une mémoire de masse annexe (le Microdrive) et des interfaces pour cartouches de ROM supplémentaire et pour l'adjonction de manettes de jeux.



Moniteur TV

Le Spectrum ne comporte pas de sortie moniteur; la qualité graphique n'est donc pas élevée. Voici le Profeel de Sony, un système de télévision modulaire avec récepteur et moniteur séparés.

Imprimante série

Outre l'imprimante du ZX, il est possible d'ajouter une imprimante conventionnelle à matrice de points ou à marguerite, via l'interface RS-232 ou la carte Centronics. L'interface 1 de Sinclair accepte les périphériques classiques.

Synthèse musicale

Le Trichord de Petron est un bon exemple de synthétiseur musical utilisable avec le Spectrum. Il est susceptible d'offrir 6 134 accords lorsqu'il est utilisé avec un Spectrum de 48 K.

Unité de cassettes

Même avec le Microdrive enfiché, le Spectrum peut encore communiquer avec une unité de cassettes standard, et ainsi utiliser des logiciels sur cassette.

Synthèse vocale

« Cheetah Sweet Talker » est un synthétiseur utilisant le système phonatoire simplifié pour produire de la parole décomposée en syllabes. 63 phonèmes et 4 types d'intervalles de diverses longueurs sont disponibles.

Interface 2

L'interface Sinclair pour manette de jeux comporte également un connecteur ROM qui facilite l'implantation de jeux ou de langages divers.

Clavier

Bien que le clavier du Spectrum représente un progrès sur ceux du ZX-80 et du ZX-81, il est toujours peu pratique pour la frappe.

Interface 1

Conçu spécialement pour le Microdrive, l'interface 1 concerne également d'autres périphériques, tels que imprimante, traceur ou modem. Il permet également de constituer un réseau local entre plusieurs Spectrum.

Imprimante ZX

Imprimante bon marché pour usage domestique, destinée d'abord à lister des programmes. Elle est de type à « impression par brûlure ».

RPP Connexion

Les réseaux de portes programmables (ULA) assurent la totalité des fonctions d'un ordinateur personnel, sauf celles de l'UC, des ROM et des RAM.

Le composant appelé « réseau de portes programmables » (Uncommitted Logic Array) est l'un des résultats les plus significatifs de la recherche en électronique qui accompagna le développement des micro-ordinateurs. Cette tranquille révolution, passée inaperçue du grand public, a tellement réduit l'encombrement des composants qu'il est devenu possible de construire des micro-ordinateurs très complexes autour de quatre composants fonctionnalisés seulement : une unité centrale, des mémoires morte et vive, et, pour relier le tout, un réseau de portes programmables (RPP).

De quoi s'agit-il ? De portes logiques non programmées initialement et susceptibles d'être programmées au fur et à mesure des besoins de fonctionnement. En ce sens le RPP s'apparente à la ROM car son contenu ne peut être modifié par l'utilisateur.

Avant sa programmation, le RPP est constitué simplement d'un grand nombre de cellules non connectées réparties en couches de composants semi-conducteurs. La dernière couche est conductrice et assure les connexions entre cellules. C'est le grand nombre d'interconnexions possibles qui donne sa flexibilité au RPP. Bien

que chaque cellule soit assez simple (quelques transistors ou une seule résistance), elles peuvent être connectées entre elles par la couche finale pour constituer des circuits complexes tels que les bascules.

De tels circuits, appelés « modules », peuvent être constitués de moins d'une douzaine de cellules, mais d'autres comportent jusqu'à plusieurs centaines de cellules. Interconnectés entre eux, ces modules constituent des circuits complexes (registres, compteurs, circuits de synchronisation). Les fonctions effectuées par ces circuits sont généralement assurées, dans un micro-ordinateur, par un certain nombre de composants logiques fonctionnalisés.

Un RPP peut être programmé pour effectuer une gamme très variée de tâches : synthétiser du son ; contrôler l'exposition, la mise au point et le moteur d'une caméra ; assurer l'affichage digital de la température d'un thermomètre. En dehors du RPP, peu de circuits sont nécessaires : une batterie électrique, un interrupteur, un capteur et des touches de commande.

L'ordinateur lui-même intervient dans la conception des couches d'interconnexion des cellules. Le mini-ordinateur DEC PCP11/23 gère un système de CAO (conception assistée par ordinateur). Il élabore en premier lieu un diagramme codé du circuit logique désiré. Le système dessine et code une carte selon la disposition correspondante. Cela a lieu sur un écran graphique avec éventuellement une copie papier obtenue sur table traçante.

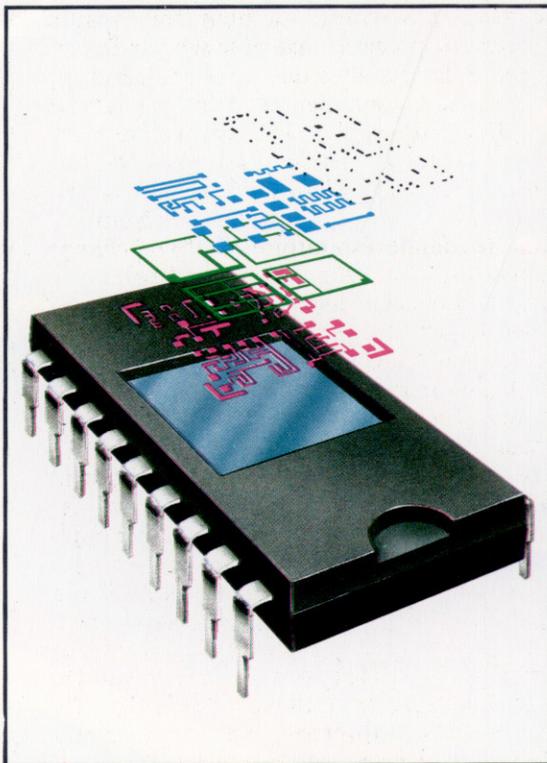
Une fois la conception effectuée, le plan est transmis à un gros ordinateur qui vérifie la conformité avec le circuit logique d'origine, et s'assure qu'il ne comporte pas d'erreur. Il est alors transmis à un autre programme qui simule à partir d'un programme test le circuit devant résulter. L'ordinateur dispose alors du savoir-faire pour obtenir le masque optique nécessaire pour fabriquer la couche finale.

Jusqu'où les réseaux de portes programmables peuvent-ils aller ? L'idée d'implanter une grande quantité de circuits de silicium simples et de laisser à l'utilisateur la possibilité de décider de leur interconnexion est certes attrayante. Cependant, au niveau actuel de la technologie, les RPP ne sont rentables que lorsqu'il existe au moins une demande commerciale pour quelques milliers de circuits identiques.

Les mémoires mortes programmables (PROM), les mémoires mortes reprogrammables (EPROM), les mémoires mortes reprogrammables électriquement (EEPROM) et les mémoires mortes modifiables électriquement (EAROM) sont autant de solutions de rechange à la mémoire morte, susceptibles d'être programmées par l'utilisateur équipé de manière adéquate. Mais il se peut qu'apparaissent bientôt des équivalents programmables par l'utilisateur des réseaux de portes programmables.

Plan supérieur

Les composants sont constitués de couches de semi-conducteurs gravées indépendamment pour créer les éléments de circuit. La dernière couche, ou plan supérieur du composant, détermine les interconnexions. Un RPP consiste en un réseau d'éléments logiques susceptibles d'être combinés entre eux pour former un circuit logique complexe. (Cl. Steve Cross.)





Faire semblant

La simulation est un excellent exercice que les ordinateurs familiaux permettent. Regardons quelques programmes disponibles sur le marché.

Les vents

Ce programme de simulation, disponible sur le BBC Micro, vous met dans la situation d'un capitaine d'un vieux voilier. Sur l'écran, une carte du monde se détache et votre bateau est représenté par un petit point. Il s'agit de barrer en utilisant les moyens du bord. Vous choisissez vos ports de départ et d'arrivée et les dates du voyage. La vitesse du bateau dépend des prévisions météo relatives aux directions et à la force des vents qui apparaissent en bas de l'écran. La position du bateau, exprimée en termes de longitude et de latitude, s'inscrit sur l'image ainsi que les zones de vents particuliers, la date, la distance parcourue et la longueur totale du voyage.

Prenons le chemin le plus court entre Le Havre et Rio de Janeiro et décidons de partir un 1^{er} janvier. Avant d'atteindre l'équateur, nous avançons rapidement en profitant des vents d'ouest. Puis c'est pratiquement le surplace pendant trois jours au passage du « Pot au Noir ». Éventuellement, un vent du sud-ouest se lève, mais cela pose un problème : comment peut-on voguer vers le sud-ouest avec des vents qui viennent de cette direction ? La solution est de faire des zigzags, sud puis ouest et ainsi de suite, jusqu'à Rio, soit 15 400 km et 205 jours plus tard.

D'autres voyages sont plus risqués : les tornades et autres icebergs font partie des dangers que vous devez affronter. Ce programme de simulation peut être pris de différentes manières. Pour les plus jeunes, il s'agira d'un simple jeu. Dans le cadre d'une leçon de géographie, les étudiants chercheront les différentes zones de vents et les routes maritimes les plus rapides.

Simulateur de vol

Il s'agit ici d'une version simplifiée d'un jeu d'arcades dans lequel vous êtes le pilote d'un petit avion. L'écran affiche la vue que vous avez depuis le cockpit et les instruments de bord. Des cadrans, des jauges et des lumières sont correctement visibles de façon à pouvoir voler sans difficultés. Le contrôle de l'avion est obtenu à partir de quatre touches fléchées et de quelques autres qui agissent sur les contrôles de puissance, le manche à balai, etc.

Contrairement au jeu d'arcades, vous avez ici la chance de vous familiariser avec tous

ces contrôles en partant directement dans les airs. Pour voir où se trouve l'avion, vous pouvez afficher une carte indiquant votre position, les balises de navigation et les routes de vol. Le meilleur moyen pour naviguer est de se « bloquer » sur certaines balises et de faire en sorte que l'avion soit parfaitement en ligne avec elles. Des points lumineux scintillent sur l'écran jusqu'à ce que le cadran vous indique que vous êtes dans la bonne voie. Une fois arrivé à ce stade, vous foncez droit sur la balise. Avec cette méthode, pas de problème pour rejoindre un aéroport. L'atterrissage est la manœuvre la plus délicate. Il vous faut être dans le bon couloir de vol avec les bonnes vitesse et hauteur, et le bon angle d'approche. Même si tout cela est correct, attention à l'erreur fatale : n'oubliez pas de sortir le train d'atterrissage !

Simulation physiologique

Dans ce programme, vous agissez sur un cerveau humain et votre objectif est de maintenir le corps en vie au moins cinquante minutes ! Le programme simule certaines variations physiologiques (température du corps, transpiration, etc.), que le corps connaît au cours de diverses activités. La première chose à faire est de préciser l'âge et le sexe de la personne et les activités que vous souhaitez lui voir faire. Le sommeil est l'activité la plus simple à simuler, car le corps utilise alors un minimum d'énergie. Mais vous pouvez choisir la course, l'escalade ou la marche.

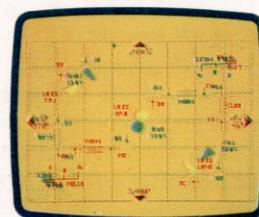
Les paramètres que vous devez contrôler sont : le volume respiratoire, le taux d'échange respiratoire et celui de la transpiration. Ensuite vous choisissez quelques valeurs initiales — par exemple un volume respiratoire de 2,5 l, un taux de quinze respirations par minute et un taux de transpiration de 3 g par minute. La simulation peut commencer.

Sur l'écran, cinq graphes correspondant aux diverses fonctions du corps apparaissent ainsi qu'une horloge. Le temps passant, les graphes montrent comment le corps réagit suivant les activités choisies et vous devez prévenir tout accident qui pourrait survenir. Si, par exemple, la température du corps monte trop, vous pouvez arrêter rapidement l'activité et accroître la transpiration pour la faire baisser. Si vous n'avez pas de chance et qu'un graphe dépasse la ligne autorisée, vous verrez un bref diagnostic : « La personne est décédée. »

Les vents



Simulateur de vol



Simulation physiologique



Ian McKinnell



Memotech MTX 512

Une conception très évoluée et des logiciels originaux de gestion du code machine sont les caractéristiques principales de cet ordinateur.

Le MTX 512 arrive très près des spécifications du fameux standard « MSX » dont nous avons déjà parlé. Et si ce n'était le composant « son » 76489 de Texas Instruments (MSX indique le composant General Instruments AY-3-8910), il pourrait revendiquer l'appellation « machine de référence ». Il diffère cependant du standard MSX par son unité centrale « Z80 », par un processeur d'affichage vidéo Texas Instruments TMS9918/9928, et un BASIC proche de la version Microsoft.

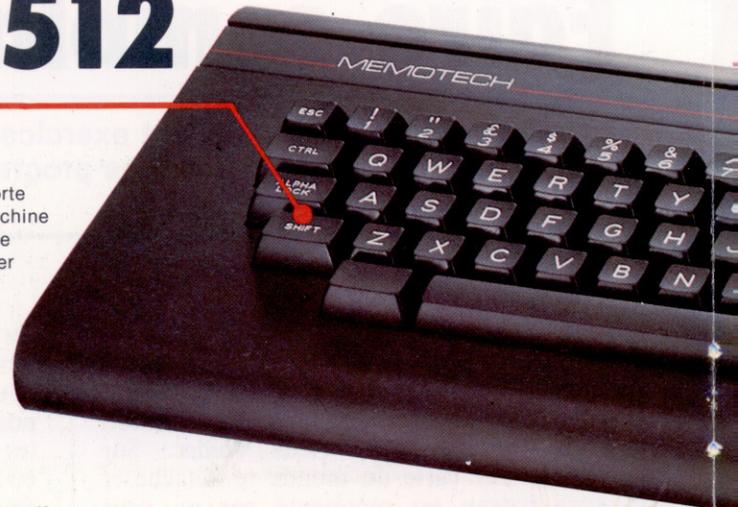
Le Memotech MTX 512 se présente comme le résultat d'une véritable synthèse technologique et d'une recherche poussée sur la forme extérieure. Ce n'est pas un de ces micros qui logent de l'électronique compliquée dans une structure hasardeuse. Son boîtier est de conception parfaite : aluminium noir et formes profilées en un seul bloc. L'accès à l'intérieur de l'ordinateur est aisé (en dévissant simplement deux boulons à clef Allen et en défaisant le fond). En comparaison avec d'autres machines, le MTX 512 a relativement beaucoup de composants. Cette solution a manifestement été privilégiée pour des raisons économiques par rapport à des réseaux de portes programmables (RPP) très importants. La disposition traditionnelle, adoptée ici, de très nombreux composants serrés les uns contre les autres permet un diagnostic de pannes plus rapide et plus aisé. Les pannes sont en effet difficilement décelables avec un RPP, et sont impossibles à réparer.

Le manuel utilisateur n'est pas des meilleurs. En dehors de sa couverture, il ne comporte pas de couleurs et les titres ne « ressortent » pas. L'absence d'index ajoute à sa difficulté d'emploi. Il est cependant très compréhensible. Memotech a décidé d'ouvrir sa machine, considérant certainement qu'il n'y avait aucun secret à l'intérieur. Tous les détails apparaissent : tables mémoire complètes, table des emplacements mémoire utiles, adressages d'entrée/sortie, diagramme du circuit imprimé, et introduction substantielle au BASIC. Il traite également du langage NODDY (voir encadré), de l'assembleur/désassembleur et du graphique.

Le MTX 512 est original en ce qu'il possède un assembleur/désassembleur fourni avec le logiciel de base, capable de programmation en code machine. Le progiciel d'assemblage ne peut pas, cependant, manipuler des adresses symboliques et des étiquettes; mais il peut être utile pour des programmes de taille moyenne, pourvu que l'on prenne des notes pendant la

Clavier

C'est un des meilleurs claviers d'ordinateur familial qui existent actuellement. Il comporte 79 touches de type machine à écrire portées par une feuille d'acier. Le boîtier d'aluminium donne à l'ensemble une bonne assise.



Interface cassette

Connecteur manettes de jeux

Deux ports sont prévus pour toutes les manettes de jeux utilisant le standard Atari.

Extensions

Le Memotech MTX 512 est conçu pour recevoir de nombreuses extensions. En particulier une carte extension mémoire et une carte double interface série avec deux ports RS232. Ceux-ci peuvent être utilisés pour les communications série normales, ou à l'aide d'un logiciel approprié, comme éléments d'un réseau spécialisé de communications.

Composant rythmeur

Le CTC Z80 assure toutes les fonctions de synchronisation du microprocesseur.

Unité centrale

Le microprocesseur est rythmé à 4 MHz.

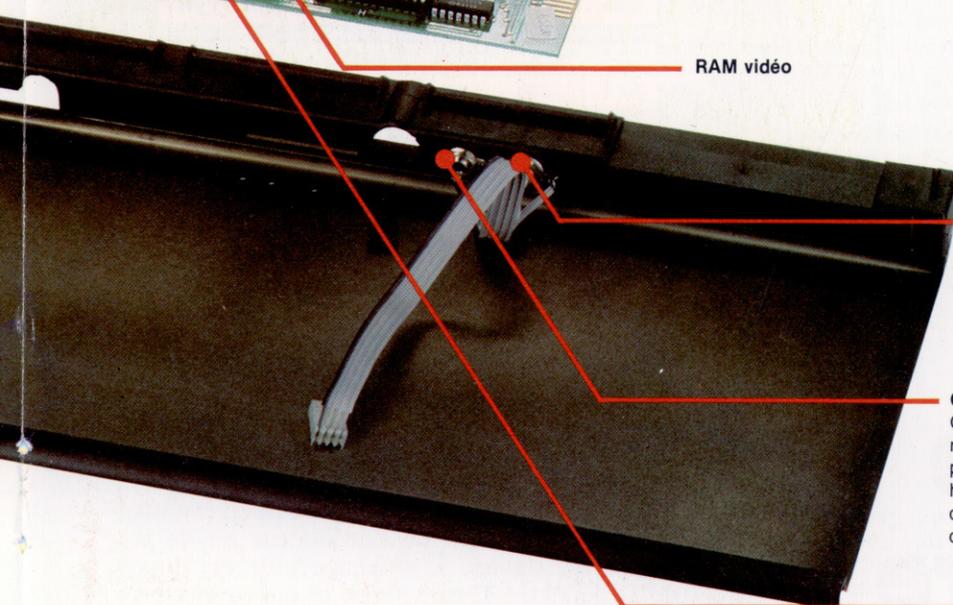
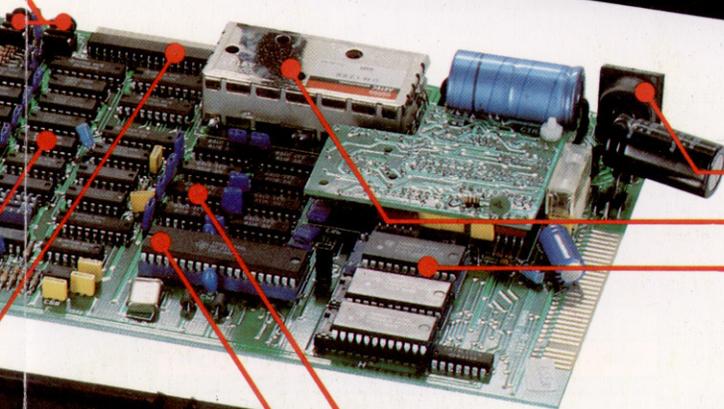
RAM utilisateur

Le MTX 512 est fourni avec 64 K de mémoire standard. Le MTX 500 dispose de 32 K.

Interface parallèle

Ce port correspond au standard Centronics pour interfaces parallèles, et également aux interfaces RS232, ce qui permet au MTX 512 d'adresser virtuellement toute imprimante.

programmation. Le logiciel de base, dit tableau de commande, est une innovation pour ce genre de machine, capable de dépistage d'erreurs code machine. Malheureusement, cette possibilité très étonnante pour une telle catégorie d'ordinateur n'a pas reçu la documentation qu'elle méritait. En effet, bien que les diverses commandes soient indiquées, leurs fonctions sont mal préci-



Connecteur alimentation

Modulateur fréquence radio

ROM

RAM vidéo

Prise moniteur

Connecteur Hi-Fi

Cet interface est rarement rencontré sur un micro. Il permet la sortie son vers un haut-parleur Hi-Fi, et donc d'obtenir une meilleure qualité.

MEMOTECH MTX 512

PRIX

★★

DIMENSIONS

488 x 202 x 56 mm.

UC

Z80.

HORLOGE

4 MHz.

MÉMOIRE

ROM : 24 K.
RAM : 64 K utilisateur,
plus 16 K vidéo
extensible à 512 K.

AFFICHAGE VIDÉO

24 lignes de 40 caractères ;
16 couleurs avec couleurs de
fond et d'écriture
indépendantes ; 127 caractères
prédéfinis, et 127 caractères à
définir par l'utilisateur.

INTERFACES

Cassette, TV, moniteur vidéo
polyvalent.

LANGAGES FOURNIS

BASIC, NODDY, ASSEMBLEUR.

AUTRES LANGAGES

En préparation.

FOURNITURES

Manuel d'installation et manuel
BASIC. Câble TV.

CLAVIER

79 touches de haute qualité.

DOCUMENTATION

Sérieuse et relativement
complète. Mais peu
mobilisatrice. Elle est
accompagnée de suffisamment
d'informations pour permettre
aux programmeurs compétents
de maîtriser à fond la machine.

Composant graphique

C'est un TMS 9928 de
Texas Instruments. Il
contrôle l'émission vidéo et
donne au MTX un graphique
similaire à celui du T199/4A
et du SORD M5. Cependant,
le MTX dispose également
de ses propres ressources
graphiques, comme la
possibilité de diviser l'écran
en plusieurs fenêtres.

NODDY

NODDY est un langage fourni avec la disquette système. Il apporte une dimension unique à la machine. Langage de débutants apprenant par eux-mêmes, NODDY est un langage simple mais finalement très sophistiqué pour certaines commandes. Limité à 11 commandes, il ne traite pas de chiffres. En effet, ce langage privilégie l'information textuelle comme base de données.

sées et il est donné trop peu d'exemples pour pouvoir espérer comprendre leur utilisation.

Le MTX 512 de Memotech peut se voir adjoindre un grand nombre d'extensions qui en feront un micro puissant. Beaucoup d'utilisateurs seront conquis et de nombreux logiciels intéressants ne manqueront pas de voir le jour prochainement sur cette machine.



Son spectaculaire

Oric-1 permet un contrôle du son pour un petit budget.

De toutes les ressources de l'Oric-1, les capacités sonores apparaissent réellement les plus impressionnantes : sept octaves, trois oscillateurs, un générateur de bruits et sept niveaux sonores prédéterminés pour choisir les sons produits. La sortie son se fait par l'intermédiaire du haut-parleur du téléviseur.

Le BASIC propre à Oric-1 comporte un jeu de commandes de son — ZAP, PING, SHOOT et EXPLODE — dont les noms parlent d'eux-mêmes. Le programme suivant vous montre comment les utiliser, et met en œuvre la précieuse commande WAIT / pause qui entraîne l'arrêt de l'ordinateur pendant quelques centièmes de seconde (ici, deux secondes).

```
10 ZAP : WAIT 200
20 EXPLODE
30 GOTO 10
```

La commande SOUND convient mieux à vos propres effets sonores :

SOUND C,P,V

Où C = Channel / Canal ou numéro de l'oscillateur (1-6), P = Pitch / Ton (10-5000), et V = Volume (0-15). C choisit le canal : 1, 2, 3, respectivement pour les oscillateurs 1, 2 et 3 (4, 5, 6 sont équivalents mais avec la présence d'un bruit). Le ton n'est pas très précis mais 10 est la note la plus haute (à 10 KHz) et 5000 est la note la plus basse (à 100 Hz). Le volume sonore maximal est 15, mais 6 convient le mieux. Lorsque V est à 0, le contrôle du volume est assuré

par le niveau sonore (Envelope) choisi par la commande PLAY.

L'inconvénient majeur de la commande SOUND est l'absence de contrôle de la durée d'une note, ce qui signifie également que SOUND ne peut s'interrompre de lui-même. Pour arrêter une note, il faut utiliser PLAY et l'interrompre en spécifiant tous les chiffres à 0.

La commande MUSIC sert à préciser une note. La syntaxe de cette commande permet, par sa simplicité, d'accéder à des programmes musicaux complexes :

MUSIC C,O,N,V

Où C = Channel/canal (1, 2 ou 3), O = Octave (0-6), N = Note (1-12), et V = Volume (0-15). Cette commande fonctionne de manière semblable à SOUND. C choisit l'oscillateur (1, 2 ou 3). Le mixage d'un bruit à un oscillateur n'est pas possible avec MUSIC. Le volume varie de 0 (le contrôle passe alors à la commande PLAY), jusqu'à 15. Octave permet de retenir une octave déterminée pour la note. Octave à 0 permet l'obtention des notes les plus basses (à partir de 32,7 Hz). L'octave 6 atteint 3951,07 KHz. Pour la partie N (note) de la commande, les nombres de 1 à 12 correspondent aux notes habituelles de la gamme :

1	2	3	4	5	6
do	do #	ré	ré #	mi	fa
7	8	9	10	11	12
fa #	sol	sol #	la	la #	si

MUSIC et PLAY

Le programme suivant utilise les commandes MUSIC et PLAY pour l'accord do, sol, mi en do majeur, successivement avec tous les niveaux sonores :

```
10 REM *****
20 REM *ACCORD*
30 REM *****
40 MUSIC 1,4,1,0 : REM
*DO*
50 MUSIC 2,3,8,0 : REM
*SOL*
60 MUSIC 2,3,5,0 : REM
*MI*
70 FOR MI = 1 TO 7 : REM
*SELECT ENV*
80 PLAY 7,0,4 MI, 750 : REM
*PLAY ACCORD*
90 PLAY 0,0,0,0 : REM
*STOP ACCORD*
100 WAIT 50 : REM
*PAUSE*
110 NEXT MI : REM *NEXT ENV*
```

Jeu de lumière

Les possibilités graphiques du Spectrum sont faciles d'emploi.

Le Spectrum est un excellent point de départ pour s'intéresser au graphisme couleur de haute résolution. Sa simplicité le rend accessible aux programmeurs débutants.

À côté des majuscules et des minuscules, il existe des caractères propres au Sinclair. L'affichage peut se faire en huit couleurs. Les com-

mandes INK, PAPER et BORDER déterminent respectivement la couleur d'écriture, de fond et de marge. En dehors du jeu normal de caractères, il est possible de définir jusqu'à 21 caractères graphiques.

L'affichage est de 24 lignes de 32 caractères. Les deux rangées inférieures sont réservées aux messages provenant de l'ordinateur ou de la frappe au clavier. L'affichage utile est donc de 22 lignes de 32 caractères, ce qui équivaut à 176 x 256 points (pixels) en haute résolution. L'une des caractéristiques les plus utiles du Spectrum est sa capacité de mélanger sur l'écran affichage haute résolution et texte. Cela permet la création de diagrammes légendés, de cartes à bâtonnets, etc. Une page écran peut être sauvegardée sur bande. Elle sera rappelée ultérieurement si besoin est. La commande SCREEN est responsable de cette opération ; elle peut également effectuer une copie d'écran sur imprimante.

La sortie basse résolution est également possible par la commande PRINT AT. Elle permet de



Pour jouer un *fa* (note 6) à 440 Hz, sur le canal 1, à un volume sonore de 6, la commande sera :

MUSIC 1,3,10,6

Cependant, pour profiter au mieux des ressources de l'Oric-1, il est préférable d'utiliser MUSIC, conjointement à PLAY dont voici la syntaxe :

PLAY C,N,E,P

Où C = Channel/canal (0-7), N = Noise/bruit (0-7), E = Enveloppe/niveau sonore (1-7), P = Enveloppe period/durée des notes de niveau sonore E (0-32767).

Les commandes Channel/canal et Noise/bruit correspondent à des options beaucoup plus complexes :

Numéro	Canal	Bruit
0	aucun Osc.	néant
1	Osc. 1	+ Osc. 1
2	Osc. 2	+ Osc. 2
3	Osc. 1 et 2	+ Osc. 1 ou 2
4	Osc. 3	+ Osc. 3
5	Osc. 1 et 3	+ Osc. 1 ou 3
6	Osc. 2 et 3	+ Osc. 2 ou 3
7	Osc. 1, 2 et 3	+ Osc. 1, 2 ou 3

Des commandes MUSIC (ou SOUND) définies préalablement et dont « Volume » est à 0 peuvent être combinées (ordres musicaux joués ensemble) par la commande PLAY, selon le numéro de canal choisi, pour construire des accords de trois notes maximum. La partie « Noise » (bruit) de la commande détermine la présence de bruit pour le(s) oscillateur(s). « Enveloppe » choisit l'un des sept niveaux sonores pour le volume, et est destiné à la note ou aux notes spécifiée(s). Ces options figurent au manuel utilisateur d'Oric. La seule intervention du programmeur sur E consiste à déterminer la durée des notes de niveau sonore E, à l'aide de P compris entre 0 et 32767. P varie avec E, mais à titre indicatif, un niveau sonore portant sur une note de fréquence 5000 dure approximativement 2 secondes.

Les commandes son de Oric-1 sont faciles à utiliser et révèlent de la part de leurs concepteurs une grande maîtrise de conception. Le seul autre micro-ordinateur comparable est le BBC, à la différence près qu'il va encore plus loin dans la mesure où il est capable de créer des sons. Oric-1 n'en reste pas moins, pour un petit budget, un micro musical complet.

Sound

Ce court programme utilise SOUND pour simuler le bruit d'un vaisseau spatial.

```
10 REM*****
20 REM *ATTERRISSAGE*
30 REM
40 FOR P = 10 TO 3000
STEP 10
50 SOUND 2,P,6
60 PLAY 2,0,1,1
70 NEXT P
80 WAIT 75
90 PLAY 0,0,0,0
100 END
```

Un sourire...

Ce programme illustre l'utilisation des commandes PLOT, CIRCLE et DRAW par la création d'un visage souriant.

```
10 REM *UN SOURIRE...*
20 CLS
30 BORDER 6
40 PAPER 6
50 INK 2
60 CIRCLE 122,88,50
70 CIRCLE 97,108,5
80 CIRCLE 147,108,5
90 PLOT 92,68
100 DRAW 152,68,PI/3
110 END
```

positionner un caractère aussi bien verticalement qu'horizontalement. Il existe aussi des effets spéciaux. En dehors de l'habituel mode d'affichage inversé, les modes brillance/BRIGHT et scintillement/FLASHING sont possibles. Une autre commande basse résolution de grande utilité est la commande OVER, qui permet la superposition de deux caractères. Cela est très utile pour fusionner du texte et un affichage haute résolution ou, par exemple, pour écrire sur un diagramme. OVER doit cependant être utilisé avec précaution dans le cas d'une coloration d'une partie de l'affichage par la commande INK : l'affichage antérieur change alors aussi.

L'affichage relève de deux zones mémoire : la première affiche les caractères et l'autre détient l'information sur les attributs des positions à l'écran. Les attributs comprennent des informations comme : les couleurs pour INK et PAPER, le mode d'affichage (FLASH, BRIGHT...), etc. Ils figurent sur un seul octet. La commande ATTR dans un programme BASIC donnera l'état des positions à l'écran.

Les commandes BASIC permettent d'accéder aisément, sur cet ordinateur, à l'affichage haute résolution. Le fait qu'il n'y ait pas d'écran haute résolution séparé permet de mélanger texte et affichage graphique.

Parmi les commandes BASIC :

PLOT x,y

attribue la couleur courante de INK au point de coordonnées (x,y).

DRAW x,y,p

trace une ligne entre la position courante du curseur et le point de coordonnées (x,y). Si l'on ajoute un troisième nombre, la ligne devient un arc de cercle. Ce troisième nombre sera une fraction de PI (la lettre π s'écrit en majuscule dans le listage). S'il s'agit de PI, l'arc de cercle tracé sera un demi-cercle. PI/2 tracera un arc plus court. Les arcs peuvent être dirigés d'un côté ou de l'autre de la ligne en donnant un signe + ou un signe - au troisième nombre.

CIRCLE x,y,r

trace un cercle de centre (x,y), et de rayon r. La plupart des commandes CIRCLE en BASIC permettent d'aplatir le cercle pour former une ellipse; mais ce n'est malheureusement pas possible avec le Spectrum.

Il existe un inconvénient majeur à utiliser les couleurs pour l'affichage haute résolution. En effet, en contrepartie de la possibilité de mélanger texte et graphique, une seule couleur pour INK ne peut être spécifiée par carré de huit pixels sur huit. Aussi, lorsque deux lignes de couleur différente se croisent, tous les pixels de la zone prendront la dernière couleur attribuée à INK.



Sixième sens

Des capteurs de lumière ou de température peuvent être connectés à un ordinateur familial. L'information sert à contrôler le chauffage central ou le système d'alarme.

Les « puces » sont de plus en plus présentes dans les appareils domestiques, comme les machines à laver, les grille-pain, les magnétoscopes ou les chaudières de chauffage. Pas étonnant, dans ces conditions, de chercher à réunir ces composants de manière à assurer leur contrôle à partir d'un système centralisé. Il est parfaitement possible de mettre au point un système organisant le fonctionnement des appareils domestiques, et cela de trois manières : par des systèmes spécialisés, par des systèmes d'alarme et par des réseaux de contrôleurs. En quoi sont-ils différents ?

Les systèmes spécialisés sont vendus sur le marché, mais il vous est possible d'en construire vous-même. Ils sont connectés à un ordinateur familial classique par des interfaces particulières. Ils règlent directement les lumières ou les thermostats. Mais pour assembler vous-même ces systèmes, vous devez posséder de bonnes connaissances en électricité et être capable d'écrire des programmes de contrôle. Les limites de ces systèmes spécialisés apparaissent dans le fait que le programme ne doit pas être interrompu. Un arrêt du courant et vos appareils sont hors circuit, comme figés, incapables de reprendre une situation correcte à l'intérieur du programme de contrôle.

Les systèmes d'alarme consistent à faire émettre un signal électronique par vos appareils — chauffage central, système antivols, détecteurs de feu ou de fumée. Quand un de ces appareils se trouve dans une situation anormale, il envoie un signal qui interrompt le programme de contrôle. Un tel système d'ensemble doit fonctionner continuellement. Mais en cas d'arrêt de la puissance électrique, il n'est pas complètement inefficace dans la mesure où les appareils sous contrôle sont partiellement autorégulés.

L'ordinateur gère plusieurs programmes dans sa mémoire : un pour chaque appareil qui lui est connecté. Disons, par exemple, que vous êtes en train de participer à un jeu d'aventures et que le détecteur de fumée envoie un message d'alarme. En réponse au signal, l'ordinateur arrête le jeu (en sauvegardant les informations relatives à l'état du jeu), et lance le programme de détection de fumée. Un message peut apparaître sur l'écran annonçant qu'un feu éventuel a été détecté ; plus simplement, une alarme peut être émise. Une fois la source de fumée découverte et éliminée, vous pouvez retourner à votre jeu, au point précis où vous l'aviez laissé. Dans le cas où le signal serait venu de l'horloge du

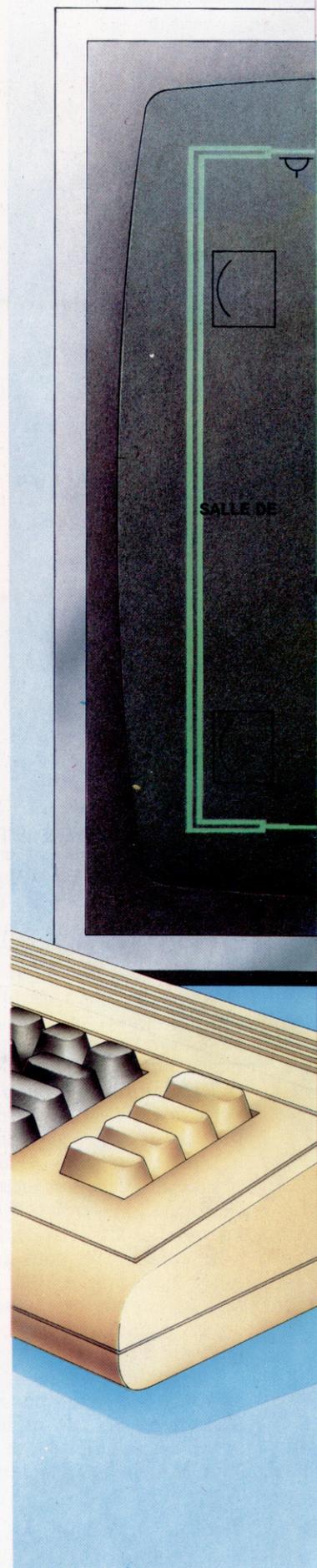
chauffage central, l'ordinateur aurait accordé l'heure avec les données fournies par les capteurs de températures (interne et externe) et réglé la chaudière en conséquence, si rapidement que vous n'auriez pas eu l'impression d'interrompre votre jeu !

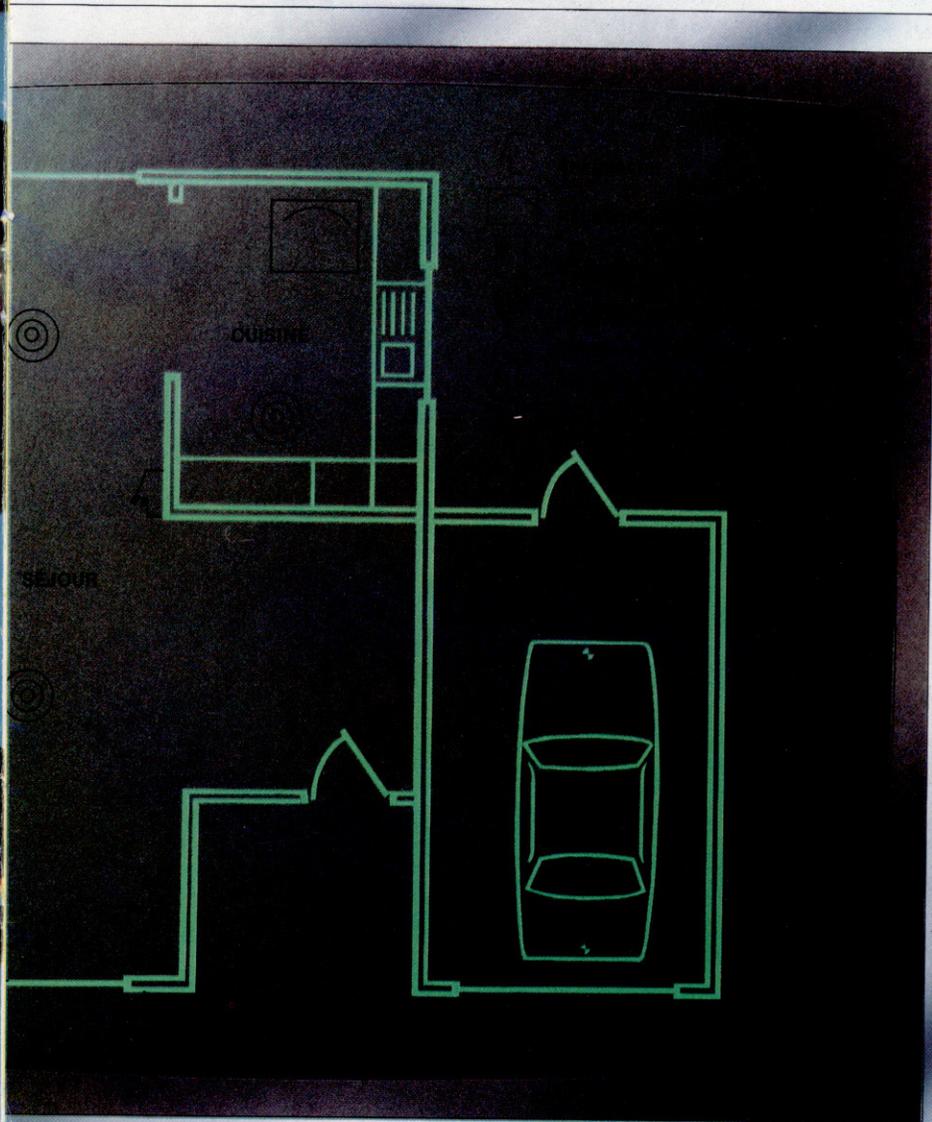
Donnons un exemple de la troisième catégorie des systèmes de contrôle des appareils domestiques (les réseaux). Un appareil astucieux utilise les principaux circuits de la maison pour contrôler toutes les unités électriques qui sont branchées. Chaque contrôleur reçoit un code numérique (une adresse), ce qui lui permet d'être mis en service ou non par l'intermédiaire d'un signal haute fréquence envoyé au travers du circuit. Les connexions nécessaires à ce type d'installation sont extrêmement dangereuses à manipuler. Seuls des électriciens qualifiés peuvent en assurer la réalisation.

Une fois un système de contrôle mis en place, il est nécessaire d'envisager l'installation d'une commande à distance (et de telle manière que l'on sache dans quel état se trouve le système). Les appareils classiques de communications comme les modems peuvent s'intégrer dans ces systèmes et permettre de faire un contrôle à partir d'un terminal à distance. Dans ce cas, une clé pour entrée dans le système est nécessaire... pour éviter que des intrus ne détournent trop facilement tout votre travail de ses objectifs.

Tout ce que nous avons décrit précédemment existe sur le marché. Les capteurs vont des simples interrupteurs des systèmes antivols aux composants électroniques très compliqués. Quelques ordinateurs domestiques peuvent sans trop de difficultés gérer de tels systèmes, mais la plupart d'entre eux ont besoin de modifications importantes.

La dépense la plus considérable pour réaliser un contrôle de votre maison à partir d'un ordinateur vient de l'achat de tout le matériel pour relier l'ordinateur aux sources d'énergie. De nombreux isolants, relais et autres commutateurs sont nécessaires pour des raisons de sécurité et de fiabilité. Mais pour l'utilisateur de ce service informatique, la tâche qui s'impose le plus est sans aucun doute l'écriture du programme. Étant donné l'importance de la rapidité des réponses que doit offrir un tel système — l'alarme doit fonctionner avant que la maison n'ait brûlé —, les programmes de contrôle doivent être écrits en langage machine. De tels programmes ne sont pas encore commercialisés ; ils le seront prochainement.

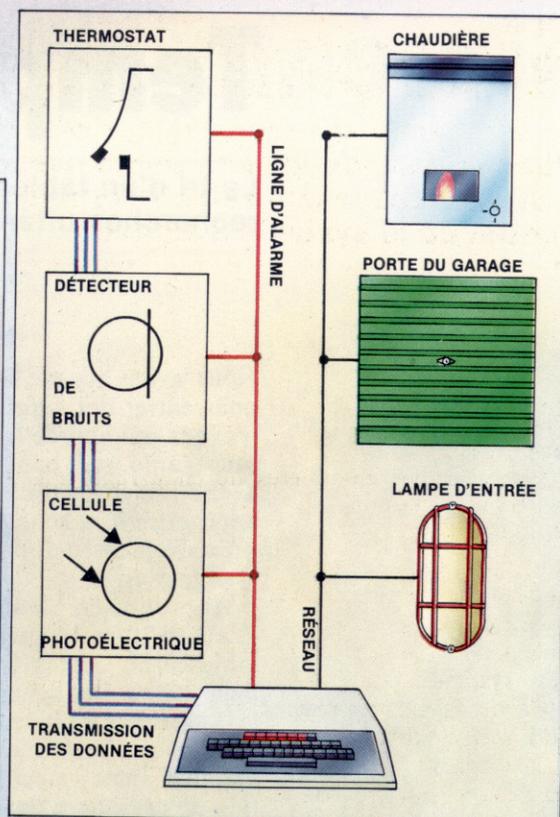




SONY

Centre de contrôle

Une représentation schématique de votre maison inscrite sur l'écran de votre ordinateur n'est pas du tout du domaine de l'impossible. Le contrôle des installations des usines et les systèmes de sécurité informatisés utilisent de telles méthodes. Naturellement, si votre système ne fonctionne que sur le principe de l'alarme, il n'est pas nécessaire de chercher à obtenir une telle image puisque votre logiciel se chargera du contrôle sans que vous ayez à vous en préoccuper. Dans peu de temps, les maisons seront conçues dès le départ avec des systèmes de surveillance informatisés. (Cl. Kevin Jones.)



Adressage

Le schéma du haut illustre deux des techniques par lesquelles l'ordinateur domestique prend en compte les appareils de la maison. Si un des trois capteurs situés à gauche enregistre une anomalie, il envoie une impulsion électronique par la ligne d'alarme. Celle-ci rejoint directement l'ordinateur, qui interrompt temporairement les programmes en cours pour sauter à un autre qui lira toutes les données arrivant du capteur par la ligne de transmission des données. Les appareils (à droite) sont reliés à un réseau de manière à ce que l'ordinateur puisse les faire réagir en leur envoyant simplement un ensemble d'informations du type : numéro de l'appareil correspondant à la porte du garage et instruction pour l'ouvrir.

Temps et mouvement

Le tri d'un tableau prend du temps en Basic, mais facilite les recherches ultérieures d'enregistrement.

Nous avons jusque-là écrit le code nécessaire pour entrer des adresses au fichier, mais nous n'avons pas abordé la délicate question de leur sauvegarde sur bande ou sur disque. Nous avons seulement oublié de créer une routine appropriée à la zone MODCHP\$ telle qu'elle a été préalablement définie dans ce cours de programmation.

Abordons-en maintenant la définition. D'abord, tous les caractères sont transcrits en majuscules, depuis la ligne 10250 jusqu'à la ligne 10330. Les lignes 10330 à 10370 comptent les caractères de la chaîne et vérifient s'ils sont des espaces. Le dernier espace trouvé donne à la variable S une valeur correspondant à la position de cet espace dans la chaîne.

Les lignes 10400 à 10420 transfèrent un par un les caractères de la chaîne des majuscules, sur CNOM\$. Les caractères sont transférés si leur valeur en code ASCII est supérieure à 64, et jusqu'au dernier espace. Cela permet d'éliminer les caractères « point » — ASCII 46, « apostrophe » — ASCII 39, « espace » — ASCII 32, et tous les autres caractères de ponctuation. Les lignes 10450 à 10470 font la même chose, à partir du dernier espace, et en opérant vers SNOM\$.

Si N\$ ne comporte qu'un seul mot, TREVANIAN par exemple, la variable \$ prendra la valeur 0, et tous les caractères seront transférés sur SNOM\$. La variable utilisée pour le prénom a été appelée CNOM\$ (Christian NAME \$/prénom \$), plutôt que FNAM\$ (Fore NAME\$/également prénom en anglais). FN pouvant prêter à confusion pour des habitués du BASIC en suggérant qu'une fonction utilisateur a été créée.

Les lignes 10490 et 10500 mettent les variables chaînes utilisées dans cette routine à 0, avant qu'elles ne soient réutilisées. Il convient de veiller à bien effectuer cette ré-initialisation pour des constructions du type $LET X\$ = X\$ + Y\$$, faute de quoi des caractères indésirables s'accumuleraient dans ces variables chaînes au cours des utilisations successives.

Remarquez que CHOI est mis à 0 dans la routine AJOUTENR afin que l'utilisateur ne mette d'enregistrement au fichier que si ce dernier est vide, à savoir uniquement à la première utilisation du programme.

Maintenant que nous pouvons ajouter autant d'enregistrements à un fichier que nous le désirons, il nous faut pouvoir sauvegarder le fichier sur disque ou sur bande. Le moyen le plus simple serait d'écrire au fichier de données (ADBK.DAT ici) les enregistrements dans l'ordre où ils ont été créés. L'inconvénient tient dans la difficulté

pour retrouver un enregistrement donné. Si nous ne sommes pas certains que tous les enregistrements du fichier ne sont pas triés, alors que l'on cherche à s'y retrouver, il faudra les passer en revue un à un, depuis le premier, pour retrouver celui désiré. Si l'enregistrement que vous recherchez se trouve être le dernier entré, chaque enregistrement du fichier demandera à être examiné avant que celui que vous désirez soit repéré.

Si le dernier enregistrement était « André Jabois » (MODCHP\$ (TAILLE 1) = « ANDRÉ JABOIS »), une routine, dont le but serait de rechercher, selon une clef, l'enregistrement, le trouverait à la condition que les enregistrements aient été préalablement triés. Il faut donc trier les enregistrements avant de les stocker, afin de ne pas perdre de temps; le tri et la recherche d'enregistrements étant très longs. Ce choix part du principe qu'il y a davantage de recherches que d'entrées (tri). Nous gardons en effet l'idée qu'un livre d'adresses est plus souvent consulté que modifié ou complété. Après utilisation du programme, les enregistrements sont triés et stockés.

Une variable RMOD est alors créée pour être utilisée comme drapeau. Elle peut prendre deux valeurs, 0 et 1. Elle est initialisée à zéro pour signifier qu'aucun enregistrement n'a été modifié pendant l'exécution du programme. Tout traitement qui modifie le fichier, comme l'addition d'un nouvel enregistrement, par exemple, met RMOD à 1. Les traitements qui ont besoin de « savoir » si le fichier a été modifié consulteront RMOD. Ainsi QUITTE, qui sauvegarde le fichier et quitte le programme, vérifie RMOD à la ligne 11050. Si RMOD est égal à 0, il n'est pas besoin de tri et de sauvegarde puisque le fichier n'a pas été modifié. D'autres routines telles que celles qui recherchent un enregistrement particulier au fichier consulteront également RMOD. Si RMOD est à 0, la recherche, ou le traitement en question, aura lieu. Lorsque RMOD=1, la routine devra, avant de pouvoir s'exécuter, faire appel à la routine de tri.

Lorsque la totalité du fichier aura été triée, la routine de tri remettra RMOD à 0.

Notre routine de tri, appelée *TRIENR* au listing du programme, remet RMOD à 0 à la ligne 11320, une fois que tous les enregistrements ont été triés. Avant d'aborder *QUITTE* (la routine qui sauvegarde le fichier et effectue END), voyons *TRIENR*. Il s'agit d'une méthode simple de tri, dite « tri par paires ou tri bulle », qui est très lente, bien qu'efficace. Un bon exemple

devrait être expliqué à partir d'une routine de tri plus efficace. Mais parmi celles qui le sont, leur compréhension deviendrait beaucoup plus difficile que celles dont nous parlons ici. Le choix de la méthode de tri, son degré de complexité et donc sa vitesse dépendent de l'importance des données à trier. Pour le tri par paires, le temps mis s'accroît selon le carré de la masse à traiter. Ainsi, si deux articles prennent deux millisecondes à être triés, quatre prendront 16 millisecondes (n^2). La progression du temps devient excessive lorsque l'on atteint un grand nombre d'enregistrements (1 000 articles, par exemple, prendraient plus de seize minutes!). Une attente de deux à trois secondes nous semble parfaitement acceptable dans le cas d'un programme comme le nôtre, mais un quart d'heure d'attente ou plus ne l'est pas. Le présent programme autorise 50 articles maximum, ce qui rend cette méthode viable. Nous indiquerons plus loin dans ce cours des méthodes de tri permettant la création dynamique de fichier qui peuvent prendre pratiquement n'importe quelle taille.

Si vous tentez une modification de ce genre dans le programme, une routine de tri plus performante serait un des premiers problèmes sur lequel il faudrait s'atteler.

Les articles triés sont les chaînes de caractères présentes en MODCHP\$(L) et MODCHP\$(L+1). L'ordre des enregistrements n'est inversé que lorsque MODCHP\$(L) est supérieur à MODCHP\$(L+1). La zone d'index (dont on ne se sert pas ici) est mise à jour aux lignes 11490 et 11570. A chaque inversion entre deux enregistrements, la variable \$ est mise à 1. Lorsque la routine de tri atteint la ligne 11290, elle vérifie la valeur de \$, et revient au début pour comparer à nouveau tous les enregistrements. Lorsque ces derniers sont en ordre, \$ garde la valeur 0 et la routine s'achève lorsque RMOD reprend la valeur 0.

La routine QUITTE (*QUITTE* au listing du programme) commence à la ligne 11000. Elle cherche d'abord si des enregistrements ont été modifiés pendant l'exécution du programme (ligne 11050 : IF RMOD=0 THEN RETURN). S'il n'y a pas eu de modification, il n'y a pas lieu de faire une nouvelle sauvegarde, QUITTE exécute alors RETURN (retour au programme principal). Cela nous ramène à la ligne 100, qui contrôle la valeur de CHOI. Si CHOI a la valeur 9 (comme cela devrait être si *QUITTE* a été exécuté), le programme principal continuera simplement jusqu'à l'énoncé FIN à la ligne 110.

Lorsque le programme trouve la valeur 1 pour RMOD à la ligne 11050, cela signifie qu'un ou plusieurs enregistrements ont été modifiés. QUITTE appelle alors la routine de tri (ligne 11070), et sauvegardera le nouveau tri sur bande ou sur disque.

La routine « sauvegarde » (*SAUVENR*) est appelée à la ligne 11090 et commence à la ligne 12000. *SAUVENR*, dans le listing principal, est écrite en BASIC Microsoft; aussi faut-il se rappeler qu'il existe des différences entre les versions du BASIC (voir Version Basic). La ligne 12030 ouvre le fichier ABDK.DAT, et donne le numéro de

canal 1 à l'opération. La ligne 12050 donne les limites pour la boucle de comptage des enregistrements du fichier. La limite supérieure est TAILLE-1, et non TAILLE, puisque la variable TAILLE a toujours une valeur supérieure d'une unité au nombre d'enregistrements du fichier (afin que lorsqu'un nouvel enregistrement est écrit, il ne le soit pas sur un ancien).

Le format des lignes 12060 et 12070 est particulièrement intéressant. Les zones sont séparées par « , », signe ou caractère prenant aussi une place au fichier. La virgule est nécessaire pour la plupart des versions BASIC car INPUT# et PRINT# s'exécutent comme INPUT et PRINT. Considérons l'énoncé INPUT X, Y, Z. Cela pourrait correspondre à une entrée à partir du clavier du genre 10, 12, 15 <CR > qui attribuerait 10, 12 et 15 à X, Y et Z respectivement. Sans les virgules l'énoncé INPUT,X,Y,Z ne saurait reconnaître les données entre elles, et assignerait les trois à la première variable. De même l'énoncé INPUT# ne saurait distinguer, sans les virgules, les enregistrements, et assignerait à chaque chaîne autant de caractères qu'elle peut en contenir. Puisque les chaînes, pour la plupart des BASIC, peuvent comporter 255 caractères, toutes les données du fichier seraient bientôt assignées bien avant que la boucle FOR L=1 TO TAILLE-1 ne soit achevée. Il en résulterait un message d'erreur INPUT PAST END (indique qu'un énoncé INPUT a été lancé alors que toutes les données étaient épuisées). Les variables chaînes (telles que NOMCHP\$(x)) en seraient également affectées (trop de données).

Lorsque tous les enregistrements ont été stockés, de L=1 TO TAILLE-1, *SAUVENR* exécute RETURN (revient à la ligne 90 du programme principal).

La ligne 100 vérifie la valeur de CHOI afin de savoir si la dernière opération était *QUITTE*. S'il s'agissait de 9 (sauvegarde et fin), le programme va à l'énoncé END à la ligne 110. Lorsque CHOI a une autre valeur, le programme revient à *CHOI* et demande à l'utilisateur de choisir une autre option.

Pour finir, mentionnons *TAILLE*. Cette routine commence à la ligne 12500. C'est une alternative à l'énoncé de la ligne 1510. Tel qu'il est, le programme nécessite une fonction de fin de fichier : IF EOF(1)=-1 THEN LET L=50. Cette condition existe d'une manière ou d'une autre pour tous les BASIC (soit une fonction spéciale de type EOF(x), soit une instruction PEEK sur une position mémoire). La routine *FTAILLE* à la ligne 12500 est une alternative dans la mesure où la fonction EOF n'est pas permise, auquel cas la ligne 1510 devrait être remplacée par GOSUB 12500.

Variantes de basic



Avant d'exécuter le programme répertoire, vous devez créer le fichier sur bande :

```
REM CRÉATION FICHIER ZONE NOM (NCHP)
20 DIM Z$(1,30)
30 LET Z$(1) = @ D'ABORD »
40 SAVE « NCHP » DATA Z$(1)
50 STOP
```

Q
R
S
T
U
V
W
X
Y
Z

SPECTRUM

A l'arrêt du programme, rembobinez la bande et tapez « NCHP » DATA Z\$(), afin de vous assurer que la commande SAVE a bien fonctionné. Cette vérification des données et des programmes sauvegardés double le temps nécessaire pour que s'arrête le lecteur de cassette, mais c'est une précaution utile avant de l'éteindre.

Voici les lignes de programme propres au Spectrum dans le listage principal :

```

1100 REM *CRETAB*
1110 DIM N$(50,30)
1120 DIM M$(50,30)
1130 DIM R$(50,30)
1140 DIM V$(50,15)
1150 DIM C$(50,15)
1160 DIM R$(50,15)
1170 DIM X$(50,30)
1180 DIM B$(30)
1190 DIM Z$(30)
1250 LET Z$ = « @PREMIER »
•
1400 REM *LECFCR* SP
1410 LOAD « NCHP » DATA N$(I)
1420 IF N$(1) = Z$ THEN LET Q$ = Z$ : RETURN
1430 LOAD « CHP » DATA M$(I)
1440 LOAD « RCHP » DATA R$(I)
1450 LOAD « VCHP » DATA V$(I)
1460 LOAD « CCHP » DATA C$(I)
1470 LOAD « TELCHP » DATA P$(I)
1480 LOAD « NDXCHP » DATA X$(I)
1490 REM FLTAILLE
1500 GOSUB 12500
•
1540 RETURN
•
1640 IF Q$ = Z$ THEN LET TAILLE = 1
•
3520 IF Q$ = Z$ THEN GOSUB 3860 : RETURN
•
3810 LET CHOI = CODE A$ - 48
•
10090 LET Q$ = « »
•
10200 REM *MODNOM* TR
•
10250 LET D$ = N$(TAILLE) : LET P$ = « »
10260 FOR L = 1 TO LEN (D$)
10270 LET A$ = D$(L)
10280 LET T = CODE A$
10290 IF T >= 97 THEN LET T = T - 32
10300 LET A$ = CHR$ T
10310 LET P$ = P$ + A$
10320 NEXT L
10330 LET D$ = P$ : LET P$ = « » : LET A$ =
« » : LET T = LEN(D$)
10340 REM TROUVER LE DERNIER ESPACE
10350 FOR L = 1 TO T
10360 IF D$(L) = « » THEN LET S = L : LET L = T
10370 NEXT L
10380 REM RETIRER LE SUPERFLU
10390 REM METTRE PRÉNOM DANS P$
10400 FOR L = 1 TO S - 1
10410 IF CODE (D$(L)) > 64 THEN LET P$ = P$ +
D$(L)
10420 NEXT L
10430 REM RETIRER L'INUTILE
10440 REM METTRE LE NOM EN A$
10450 FOR L = S + 1 TO LEN (D$)
10460 IF CODE(D$(L)) > 64 THEN LET A$ = A$ +
D$(L)
10470 NEXT L
10480 LET M$(TAILLE) = A$ + « » + P$
10490 LET P$ = « » : LET A$ = « »
10510 RETURN
•

```

N.B. — La routine ci-dessus divise le nom au premier espace, et non au dernier, du fait du traitement propre au ZX Spectrum.

```

12000 REM *SAUVENR* SP
12030 SAVE « NCHP » DATA N$(I)
12040 SAVE « MCHP » DATA M$(I)
12050 SAVE « RCHP » DATA R$(I)
12060 SAVE « VCHP » DATA V$(I)
12070 SAVE « CCHP » DATA C$(I)
12080 SAVE « TELCHP » DATA P$(I)
12090 SAVE « NDXCHP » DATA X$(I)
•

```

```

12150 RETURN
•

```

```

12500 REM *FLTAILLE*SP
12510 LET TAILLE = 50
12520 FOR L = 1 TO 50
12530 IF N$(L) = B$ THEN LET TAILLE = L : LET
L = 50
12540 NEXT L
12560 RETURN

```

LYNX

Le Lynx ne comporte pas la commande STR\$. Pour DIMensionner des tableaux chaînes, voir nos précédentes « variantes de basic ».

EOF

Pour le Commodore 64 et le Vic-20, remplacez la ligne 20 par :

```

1520 IF ST AND 64 THEN LET L = 50

```

Pour le Dragon 32, supprimez la ligne 1520 et remplacez-la par :

```

1485 IF EOF (- 1) THEN GOTO 1510

```

Sur le BBC, remplacez la ligne 1520 par :

```

1520 IF EOF # X THEN LET L = 50

```

où X est la variable numérique utilisée dans l'énoncé OPENOUT.

**OPEN
CLOSE**

```

10 REM 'PROGRAMME PRINCIPAL'
20 REM 'INITIL'
30 GOSUB 1000
40 REM *ACCUEIL*
50 GOSUB 3000
60 REM *CHOIX*
70 GOSUB 3500
80 REM *EXECUT*
90 GOSUB 4000
100 IF CHOI <> 9 THEN 60
110 END
1000 REM SOUS-PROGRAMME *INITIL*
1010 GOSUB 1100 : REM SOUS-PROGRAMME *CRETAB*
(CRÉATION TABLEAUX) SOUS-PROGRAMME
1020 GOSUB 1400 : REM SOUS-PROGRAMME *LECFCR*
(LECTURE FICHER) SOUS-PROGRAMME
1030 GOSUB 1600 : REM SOUS-PROGRAMME *DEFDRA*
(METTRE DES DRAPEAUX)
1040 REM
1050 REM
1060 REM
1070 REM
1080 REM
1090 RETURN
1100 REM SOUS-PROGRAMME *CRETAB* (CRÉATION TABLEAUX)
1110 DIM NOMCHP$(50)
1120 DIM MODCHP$(50)
1130 DIM RUECHP$(50)
1140 DIM VILCHP$(50)
1150 DIM CPOCHP$(50)
1160 DIM TELCHP$(50)
1170 DIM NDXCHP$(50)
1180 REM
1190 REM
1200 REM
1210 LET TAILLE = 0
1220 LET RMOD = 0
1230 LET SVED = 0
1240 LET ACT = 0
1250 REM
1260 REM
1270 REM
1280 REM
1290 REM
1300 RETURN
1400 REM SOUS-PROGRAMME *LECFCR*
1410 OPEN « 1 », #1, « ADBK .DAT »
1420 INPUT #1, TEST$
1430 IF TEST$ = « @PREMIER » THEN GOTO 1530 : REM FERMER
ET RETOUR
1440 LET NOMCHP$(1) = TEST$
1450 INPUT #1, MODCHP$(1), RUECHP$(1), VILCHP$(1),
CPOCHP$(1), TELCHP$(1)
1460 INPUT #1, NDXCHP$(1)
1470 LET TAILLE = 2

```



```

1480 FOR L = 2 TO 50
1490 INPUT #1, NOMCHP$(L), MODCHP$(L), RUECHP$(L),
      VLCHP$(L), CPOCHP$(L)
1500 INPUT #1, TELCHP$(L), NDXCHP$(L)
1510 LET TAILLE = TAILLE + 1
1520 IF EOF(1) = - 1 THEN LET L = 50
1530 NEXT L
1540 FERMER #1
1550 RETURN
1600 REM SOUS-PROGRAMME 'DEFDRA'
1610 REM METTRE DRAPEAUX APRES 'LECFOR'
1620 REM
1630 REM
1640 IF TEST$ = * @'PREMIER' THEN LET TAILLE = 1
1650 REM
1660 REM
1670 REM
1680 REM
1690 RETURN
3000 REM SOUS-PROGRAMME 'ACCUEIL'
3010 PRINT CHR$(12) : REM EFFACER ÉCRAN
3020 PRINT
3030 PRINT
3040 PRINT
3050 PRINT
3060 PRINT TAB(12); * 'BIENVENUE A' *
3070 PRINT TAB(9); * 'ABC MICRO' *
3080 PRINT TAB(6); * 'RÉPERTOIRE' *
3090 PRINT
3100 PRINT TAB(5); * (APPUYEZ SUR LA BARRE D'ESPACEMENT
      POUR CONTINUER) *
3110 FOR L = 1 TO 1
3120 IF INKEYS <> * * THEN L = 0
3130 NEXT L
3140 PRINT CHR$(12)
3150 RETURN
3500 REM SOUS-PROGRAMME 'CHOIX'
3510 REM
3520 IF TEST$ = * @'PREMIER' THEN GOSUB 3860 : REM
      SOUS-PROGRAMME 'PREMIER'
3530 IF TEST$ = * @'PREMIER' THEN RETURN
3540 REM 'CHMENU'
3550 PRINT CHR$(12)
3560 PRINT * CHOISIR *
3570 PRINT
3580 PRINT
3590 PRINT
3600 PRINT * 1. TROUVE L'ENREGISTREMENT (NOM) *
3610 PRINT * 2. TROUVE LES NOMS (NOMS INCOMPLETS) *
3620 PRINT * 3. TROUVE LES ENREGISTREMENTS (VILLE) *
3630 PRINT * 4. TROUVE L'ENREGISTREMENT (INITIALE) *
3640 PRINT * 5. LISTE TOUS LES ENREGISTREMENTS *
3650 PRINT * 6. AJOUTE UN NOUVEL ENREGISTREMENT *
3660 PRINT * 7. MODIFIE L'ENREGISTREMENT *
3670 PRINT * 8. SUPPRIME L'ENREGISTREMENT *
3680 PRINT * 9. FIN ET SAUVEGARDE *
3690 PRINT
3700 PRINT
3710 REM 'ENTCHOI'
3720 REM
3730 LET L = 0
3740 LET I = 0
3750 FOR L = 1 TO 1
3760 PRINT * FRAPPEZ VOTRE CHOIX (1 - 9) *
3770 FOR I = 1 TO 1
3780 LET A$ = INKEYS
3790 IF A$ = * * THEN I = 0
3800 NEXT I
3810 LET CHOI = VAL(A$)
3820 IF CHOI < 1 THEN L = 0
3830 IF CHOI > 9 THEN L = 0
3840 NEXT L
3850 RETURN
3860 REM SOUS-PROGRAMME 'PREMIER' (MESSAGE
      D'AFFICHAGE)
3870 LET CHOI = 6
3880 PRINT CHR$(12) : REM EFFACE L'ÉCRAN
3890 PRINT
3900 PRINT TAB(8); * PAS D'ENREGISTREMENT *
3910 PRINT TAB(8); * AU FICHER. COMMENCEZ *
3920 PRINT TAB(6); * PAR METTRE UN ENREGISTREMENT *
3930 PRINT
3940 PRINT TAB(5); * (APPUYEZ SUR LA BARRE D'ESPACEMENT
      POUR CONTINUER) *
3950 FOR B = 1 TO 1
3960 IF INKEYS <> * * THEN B = 0
3970 NEXT B
3980 PRINT CHR$(12) : REM EFFACE L'ÉCRAN
3990 RETURN
4000 REM SOUS-PROGRAMME 'EXECUT'
4010 REM
4020 IF CHOI = 6 THEN GOSUB 10000
4030 REM
4040 REM 1 POUR 'TROUVER'
4050 REM 2 POUR 'TRNOMS'
4060 REM 3 POUR 'TRVILLE'
4070 REM 4 POUR 'TRINITE'
4080 REM 5 POUR 'LSTENR'
4090 IF CHOI = 6 THEN GOSUB 10000
4100 REM 7 POUR 'MODENR'
4110 REM 8 POUR 'EFFENR'
4120 IF CHOI = 9 THEN GOSUB 11000
4130 REM
4140 RETURN
10000 REM SOUS-PROGRAMME 'AJOUTENR'
10010 PRINT CHR$(12) : REM EFFACE L'ÉCRAN
10020 INPUT * FRAPPEZ LE NOM *; NOMCHP$(TAILLE)
10030 INPUT * FRAPPEZ LA RUE *; RUECHP$(TAILLE)
10040 INPUT * FRAPPEZ LA VILLE *; VILCHP$(TAILLE)
10050 INPUT * FRAPPEZ LE CODE POSTAL *; CPOCHP$(TAILLE)
10060 INPUT * FRAPPEZ LE NUMÉRO DE TÉLÉPHONE *;
      TELCHP$(TAILLE)
10070 LET RMOD = 1 : REM 'ENREGISTREMENT MODIFIÉ'
      DRAPEAU ARBORÉ
10080 LET NDXCHP$(TAILLE) = STR$(TAILLE)
10090 LET TEST$ = * *
10100 GOSUB 10200 : REM 'MODNOM'
10110 LET CHOI = 0

```

```

10120 LET TAILLE = TAILLE + 1
10130 REM
10140 REM
10150 RETURN
10200 REM ROUTINE 'MODNOM'
10210 REM TRANSCRIT LE CONTENU DE NOMCHP$ EN
      MAJUSCULES
10220 REM ÉLIMINE L'INUTILE, ET STOCKE DANS L'ORDRE :
10230 REM NOM + ESPACE PRÉNOM DANS MODCHP$
10240 REM
10250 LET N$ = NOMCHP$(TAILLE)
10260 FOR L = 1 TO LEN(N$)
10270 LET TEMP$ = MED$(N$,L,1)
10280 LET T = ASC(TEMP$)
10290 IF T >= 97 THEN T = T - 32
10300 LET TEMP$ = CHR$(T)
10310 LET P$ = P$ + TEMP$
10320 NEXT L
10330 LET N$ = P$
10340 REM LOCALISE LE DERNIER ESPACE
10350 FOR L = 1 TO LEN(N$)
10360 IF MED$(N$,L,1) = * * THEN S = L
10370 NEXT L
10380 REM ÉLIMINE L'INUTILE ET STOCKE LE PRÉNOM
10390 REM DANS CNOM$
10400 FOR I = 1 TO S - 1
10410 IF ASC(MED$(N$,L,1)) > 64 THEN CNOM$ = CNOM$ +
      MED$(N$,L,1)
10420 NEXT L
10430 REM ÉLIMINE L'INUTILE ET STOCKE LE NOM
10440 REM DANS SNOM$
10450 FOR L = S + 1 TO LEN(N$)
10460 IF ASC(MED$(N$,L,1)) > 64 THEN SNOM$ = SNOM$ +
      MED$(N$,L,1)
10470 NEXT L
10480 LET MODCHP$(TAILLE) = SNOM$ + * * * + CNOM$
10490 LET P$ = * * : LET N$ = * * : LET SNOM$ = * * :
      LET CNOM$ = * *
10500 LET P$ = * * : LET N$ = * * : LET SNOM$ = * * :
      LET CNOM$ = * *
10510 RETURN
11000 SOUS-PROGRAMME 'QUITTE'
11010 REM TRI ET SAUVEGARDE
11020 REM 'SIL Y A EU MODIFICATION
11030 D'ENREGISTREMENTS (RMOD = 1)
11040 REM
11050 IF RMOD = 0 THEN RETURN
11060 REM
11070 GOSUB 11200 : REM 'TRIENR'
11080 REM
11090 GOSUB 12000 : REM 'SAUVENR'
11100 RETURN
11200 REM SOUS-PROGRAMME 'TRIENR'
11210 REM TRI MODCHP$
11220 REM PAR ORDRE ALPHABÉTIQUE ET MET À JOUR NDXCHP
11230 REM
11240 REM
11250 LET S = 0
11260 FOR L = 1 TO TAILLE - 2
11270 IF MODCHP(L) > MODCHP(L + 1) THEN GOSUB 11350
11280 NEXT L
11290 IF S = 1 THEN 11250
11300 REM
11310 REM
11320 LET RMOD = 0 : REM RETIRE LE DRAPEAU 'ENREG.
      MODIFIÉ'
11330 REM
11340 RETURN
11350 REM SOUS-PROGRAMME 'ECHENR'
11360 LET TNOMCH$ = NOMCHP$(L)
11370 LET TMODCH$ = MODCHP$(L)
11380 LET TRUECH$ = RUECHP$(L)
11390 LET TVILCH$ = VILCHP$(L)
11400 LET TCPOCH$ = CPOCHP$(L)
11410 LET TTELCH$ = TELCHP$(L)
11420 REM
11430 LET NOMCHP$(L + 1) = NOMCHP$(L + 1)
11440 LET MODCHP$(L + 1) = MODCHP$(L + 1)
11450 LET RUECHP$(L + 1) = RUECHP$(L + 1)
11460 LET VILCHP$(L + 1) = VILCHP$(L + 1)
11470 LET CPOCHP$(L + 1) = CPOCHP$(L + 1)
11480 LET TELCHP$(L + 1) = TELCHP$(L + 1)
11490 LET LDXCHP$(L) = STR$(L)
11500 REM
11510 LET NOMCHP$(L + 1) = TNOMCHP$
11520 LET MODCHP$(L + 1) = TMODCHP$
11530 LET RUECHP$(L + 1) = TRUECHP$
11540 LET VILCHP$(L + 1) = TVILCHP$
11550 LET CPOCHP$(L + 1) = TCPOCHP$
11560 LET TELCHP$(L + 1) = TTELCHP$
11570 LET NDXCHP$(L + 1) = STR$(L + 1)
11580 LET S = 1
11590 REM
11600 RETURN
12000 REM SOUS-PROGRAMME 'SAUVENR'
12010 REM
12020 REM
12030 OPEN * 0 *, #1, * ADBK.DAT *
12040 REM
12050 FOR L = 1 TO TAILLE - 1
12060 PRINT #1, NOMCHP$(L); * *; MODCHP$(L); * *;
      RUECHP$(L); * *; VILCHP$(L)
12070 PRINT #1, CPOCHP$(L); * *; TELCHP$(L); * *; NDXCHP$(L)
12080 NEXT L
12090 REM
12100 REM
12110 REM
12120 REM
12130 FERMER #1
12140 REM
12150 RETURN
12500 REM SOUS-PROGRAMME 'FLTAILLE'
12510 IF NOMCHP$(L) = * * THEN LET L = 50
12520 IF NOMCHP$(L) = * * THEN RETURN
12530 LET TAILLE = TAILLE + 1
12540 REM
12550 REM
12560 RETURN

```

R
S
T
U
V
W
X
Y
Z

Vannevar Bush

L'analyseur différentiel

Cette machine était destinée à résoudre un grand nombre de fonctions mathématiques, en particulier les équations différentielles de second ordre. La méthode fut d'abord esquissée par lord Kelvin, qui cherchait à alimenter la sortie d'un « intégrateur » (un appareil qui calcule effectivement les surfaces sous des courbes) à partir de l'entrée d'un autre. La puissance de sortie était généralement trop faible pour jouer le rôle d'une entrée. Il fallut attendre les amplificateurs pour que cette méthode puisse être utilisée. En 1931, la machine de Bush était entièrement mécanique, avec des engrenages, des axes et des moteurs électriques. Au cours des années quarante, des analyseurs différentiels étaient construits à partir de composants électriques, mais le résultat fut une machine complexe pesant plus de 100 t. Les conditions initiales et les paramètres de contrôle étaient fournis par des bandes perforées. Cette machine fut utilisée pendant toute la Seconde Guerre mondiale pour l'analyse cryptographique et les travaux balistiques.



L'analyseur différentiel de Vannevar Bush était une calculatrice résolvant les équations différentielles.

De nombreux spécialistes prétendent que Vannevar Bush est le père des ordinateurs. Sa principale contribution au développement de la science informatique date de 1931, avec la création d'un analyseur différentiel mécanique. Cette initiative contribua peut-être à la naissance des ordinateurs numériques.

Né à Boston (Massachusetts, É.-U.) le 11 mars 1890, il va suivre les pas de son père en entreprenant des études d'ingénieur. Après son diplôme, cela le conduit d'abord à travailler, brièvement, à la Compagnie générale électrique, puis à l'université Harvard et au M.I.T. (Massachusetts Institute of Technology) pour faire de la recherche. Pendant la Première Guerre mondiale, Bush s'occupe du développement de détecteurs de sous-marins pour l'armée américaine.

La première invention de Bush fut un appareil de surveillance terrestre. Il était encore étudiant à l'époque. Le mécanisme de cet appareil, qui était suspendu entre deux roues de bicyclette, calculait la hauteur du sol au-dessus duquel il se déplaçait et traduisait graphiquement le profil terrestre. Le système comprenait un autre appareil dénommé intégrateur, car le calcul des hauteurs des différentes positions

demandait la connaissance de toutes les valeurs précédentes enregistrées au cours du déplacement.

Au MIT, Bush devint professeur d'électricité (transmission de la force) et il entreprit de résoudre un des problèmes majeurs soulevés par la fourniture de l'électricité : comment éviter les pannes totales qui surviennent à la suite d'une surdemande imprévisible ? Les équations mathématiques qui répondent à cette situation furent découvertes à la fin du XIX^e siècle par l'Écossais James Clerck Maxwell. Mais il y avait tant d'équations à résoudre simultanément que les solutions ne pouvaient être trouvées simplement à l'aide d'un crayon et de papier. Bush décida de concevoir une machine pour répondre au problème, en s'inspirant des travaux d'un autre scientifique, lord Kelvin (1824-1907) qui cherchait à prévoir mathématiquement les marées.

Au début des années vingt, Bush construit une première machine qu'il appelle le « producteur de télégraphe ». Cette machine permettait à un opérateur humain de suivre le mouvement des vagues dessinées sur un graphique (un potentiomètre qui transformait les mesures de distance en volts était utilisé). Par la suite, les signaux électriques alimentaient un wattmètre spécialement adapté (un disque tournant enregistre la quantité de puissance consommée par intégration des valeurs du courant et du voltage pour donner ainsi le « produit »).

Le succès de cette machine dans la résolution des équations simultanées donna l'idée d'en construire une autre pouvant aller encore plus loin et résoudre des équations différentielles de second ordre. Par la suite, les études de Bush le conduisirent à la mise au point, en 1931, du premier analyseur différentiel. La réussite de la machine fut telle que des modèles semblables furent construits en Europe. L'université de Pennsylvanie (É.-U.), où le premier ordinateur — l'ENIAC — naquit, en acquit une. Alors que la machine « produisant du télégraphe » avait une précision de 2 %, l'analyseur différentiel donnait des résultats avec 0,05 % de précision. Cependant, le coût de l'accroissement de précision de ces machines mécaniques augmentait d'un facteur 10 pour chaque décimale gagnée. Avec les ordinateurs numériques, le coût ne fait que doubler pour une même évolution de la précision.

Bush devint le doyen de l'école d'ingénieurs et le vice-président de l'institut Carnegie en 1939. Par la suite, il devint le président du Comité de recherche de la défense nationale et répartit les fonds destinés à la recherche scientifique. Pendant les années de guerre, il fut à ce titre responsable des recherches militaires de son pays. Il joua un rôle influent dans le lancement du projet Manhattan, qui conduisit à la création de la première bombe atomique. Bush prit sa retraite en 1955. Il mourut en 1974.

LE « GUIDE D'ACHAT » DE VOTRE ENCYCLOPÉDIE

Cher lecteur,

Vous avez entre les mains les premiers numéros d' ABC informatique, et nous vous remercions de votre achat.

Que vous soyez déjà un lecteur d'encyclopédies en fascicules, ou que celle-ci soit la première que vous envisagiez d'acquérir : dans les deux cas vous trouverez ci-dessous quelques informations et conseils pratiques qui vous aideront à constituer sans problèmes, chaque semaine, votre encyclopédie.

LES AVANTAGES DE L'ENCYCLOPÉDIE EN FASCICULES

- Chaque semaine, pour le prix d'un magazine, vous pouvez acquérir un ouvrage complet sur un sujet qui vous passionne, et votre dépense est étalée dans le temps.
- Vous disposez à chaque livraison d'une quantité de lecture raisonnable.
- Vous vous constituez un ouvrage de référence, que toute la famille peut consulter à loisir.
- Des reliures vous permettent de transformer vos fascicules en luxueux volumes qui mettront en valeur votre bibliothèque.

QUELQUES CONSEILS

- Achetez vos fascicules régulièrement chez le même marchand de journaux, vous serez mieux servi et vous faciliterez notre distribution.
- Si pour une raison quelconque — absence momentanée, vacances, fermeture annuelle de votre point de vente, etc. — vous ne pouvez vous procurer un ou plusieurs numéros, commandez-les chez votre marchand : dans les meilleurs délais, celui-ci recevra votre commande. Pendant toute la durée de la collection, nous vous garantissons la disponibilité de tous les numéros (et même après la fin de la collection, pendant au minimum 6 mois).
- Pour faciliter votre commande, rappelez à votre marchand de journaux :
 - le titre de l'encyclopédie : *Abc informatique*
 - la codification (c'est le numéro à 4 chiffres qui est indiqué dans un petit rectangle sur la première page de couverture) :

Abc informatique M. 6062

— les numéros des fascicules souhaités, par exemple :

*Abc informatique M. 6062
numéros 7, 8 et 9*

— si vous aviez malgré tout des difficultés pour trouver régulièrement votre encyclopédie à votre point de vente habituel, adressez une simple lettre à notre : Service de ventes, Éditions Atlas, 33, avenue du Maine, 75755 PARIS Cedex 15, en indiquant clairement :

- votre nom et votre adresse,
- l'adresse de votre marchand de journaux,
- la nature du problème (service irrégulier, numéros manquants, etc.).

Nous prendrons immédiatement les dispositions nécessaires pour y remédier, et vous en tiendrons informé.

LES RELIURES

Nous mettons à votre disposition des reliures qui contiennent chacune 12 numéros.

Ces reliures sont identiques pour tous les volumes de la collection. Un système de transfert, livré avec chaque reliure, permet d'indiquer sur le dos le numéro du volume. S'il arrivait que votre point de vente manque momentanément de reliures, il peut vous les commander et vous les fournir rapidement.

La codification des reliures d' ABC informatique est : M. 6103.

Passez votre commande en indiquant, par exemple :

*reliure Abc informatique M 6103
3 exemplaires*

Pour être certain de pouvoir toujours conserver vos fascicules en parfait état, ayez toujours une ou plusieurs reliures d'avance. Vous y classerez vos fascicules au fur et à mesure de leur parution.

Encore une fois, n'hésitez pas à nous contacter pour tout problème que vous pourriez rencontrer dans l'acquisition de votre encyclopédie. Nous n'en connaissons pas qui ne puissent se résoudre.

ÉDITIONS ATLAS

Protégés par une élégante reliure, vos numéros d'aBc Informatique seront plus faciles à consulter

Pour classer, répertorier, protéger vos fascicules d'ABC Informatique, les Éditions Atlas vous proposent des reliures élégantes, sobres, qui s'insèrent parfaitement dans votre bibliothèque. Chacune contient 12 fascicules,

les maintient, les préserve. Un système simple, résistant, vous permet de les assembler facilement. Elles sont en vente en permanence chez votre marchand de journaux. Demandez-les !



Chaque
reliure:
40 FF
295 FB
18 FS

Cod. N.M.P.P. :
6103

EDITIONS
ATLAS